MINIMUM WEIGHT DESIGN OF A STEEL TRUSS WITH STRESS AND DEFORMATION CONSTRAINTS, SUBJECT TO MULTIPLE LOADING CONDITIONS AND USING AVAILABLE SECTIONS

Alfreds Vikmanis

A MAJOR TECHNICAL REPORT

in

The Faculty

of

Engineering

Presented in Partial Fulfillment of the Requirements of
the Degree of Master of Engineering at
Sir George Williams University
Montreal, Canada

September, 1974

ABSTRACT

MINIMUM WEIGHT DESIGN OF A STEEL TRUSS WITH STRESS AND DEFORMATION CONSTRAINTS, SUBJECT TO MULTIPLE LOADING CONDITIONS AND USING AVAILABLE SECTIONS

by

Alfreds Vikmanis

ABSTRACT

In this study a procedure is developed for converting an existing set of unrelated computer programs in structural analysis and linear programming into an integrated design sequence for designing a statically indeterminate steel truss of minimum weight subject to stress and deformation constraints and multiple loading conditions.

It shows that with relatively little additional programming, savings in material ~~and also in the designers'~~ time can be achieved. It also demonstrates the advantages of having a small, self-contained computer program for solving a routine problem.

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

Sincere appreciation is extended to Dr. M. McC. Douglass and Dr. A.S. Ramamurthy for their help and guidance in completing this study.

It is part of an engineer's continuing education and the writer is grateful to the Faculty of Engineering, Department of Graduate Studies for making such an opportunity available.

Also, much appreciated is the generous amount of computer time provided by the Computing Center of Sir George Williams University.

TABLE OF CONTENTS

TABLE OF CONTENTS

PAGE

LIST OF TABLES AND ILLUSTRATIONS

LIST OF TABLES

LIST OF ILLUSTRATIONS

NOTATIONS

NOTATIONS

[A]	statics matrix of structural system
A_i	area of design parameter
$[B] = [A]^T$	compatibility matrix of structural system
[AE]	statics matrix of a structural element
$[BE] = [AE]^T$	compatibility matrix of a structural element
[SE]	internal stiffness matrix of a structural element
[KE]	external stiffness matrix of a structural element
E_i	modulus of elasticity of design parameter i
[F]	internal force matrix of structural system
[ΔF]	matrix of changes of internal forces due to a change in one design parameter
ΔF_{ijk}	change in internal force <u>i</u> due to a change in design parameter <u>j</u> under the action of loading condition <u>k</u>
F_a	allowable axial stress
$(f_{ik})ALL$	allowable stress of member i in loading condition k
$[K] = [ASA^T]$	external stiffness matrix of structural system
Li	length of design parameter i
NF	total number of designated internal forces
NLC	number of loading conditions
NM	number of design parameters
NP	number of external forces and deformations (number of freedoms)
[P]	external applied force matrix

[S]	internal stiffness matrix of structural system
[ΔS]	change in internal stiffness matrix due to change in one design parameter
I_i	moment of inertia of design parameter <u>i</u>
{x}	external deformation matrix of structural system
[Δx]	matrix of changes in external deformations due to change in one design parameter
Δx_{ijk}	change in deformation <u>i</u> due to a change in design parameter <u>j</u> under the action of loading condition <u>k</u>
(X_{ik})all	allowable deformation i in loading condition k
u_i	solution variable, representing multiplier in design parameter <u>i</u>
L_{Bi}	lower bound on change in solution variable u_i
U_{Bi}	upper bound on change in solution variable u_i
α_i	angle of inclination of member i
W	total weight of structural system
H_i	horizontal projection of a truss member
V_i	vertical projection of a truss member
A_i	area of a section of a truss member i (Design parameter)

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

The purpose of structural design is to create a structure which can serve its intended function well and is most economical to build and operate. For Civil Engineering structures, the weight of the structural frame is usually taken as criterion of the optimality assuming that the material has been chosen. The objective is to use as little of the material as possible and to have as low a weight as possible.

Except for statically determinate structures without limitations in deflections, structural design is generally a trial and error procedure where the designer assumes member sizes and then analyzes the structure to find the stresses and deflections. The intention is to use the material as well as possible and have every member stressed to the max. allowable stress for at least one loading condition satisfying deflection requirements at the same time.

Before computer-aided structural analysis became available, the success of a structural design was dependent solely on the designer's experience with structures of a similar type so that he was able to choose the right proportions for the members. Not many trials were possible because of time and expenses involved in re-analyzing a design. Therefore, fully stressed designs were not often achieved except for statically determinate structures and there was no way of telling if a design was optimal.

As the digital computer became available, things changed to some extent. Automatic methods of analysis were established and it was possible to do more re-analyzing cycles as computer operations became less expensive and computer capacities were enlarged. Matrix methods became practical as they can be efficiently programmed for computers. This kind of structural analysis is now accepted in design offices and programs are available for analyzing almost any kind of structure. First, structures consisting of linear elements were solved - trusses, frames, grids, beams, then plates and shells, when finite elements were introduced.

These more efficient methods of analysis still were not helping the designer in choosing the right member properties for new trials and a strong tendency remained to limit design and analysis to one or two cycles because the inter-relations between member sizes and forces in members, except for the very simplest structures are rather complex. Intuition is often not good enough as a guide for making the right choice.

The next step in the evolution of structural design was to develop procedures where new member sizes are automatically chosen such that each cycle brings the solution closer to the optimum-minimum weight of the structural frame in most cases.

Basically, these are iterative methods with changes in member size in each cycle, only now the changes are guided by mathematical programming or other means so as to achieve the optimality.

These methods are somewhat more complicated than straight analysis, they require larger computer capacities, more computer time and are not quite yet accepted by the design industry. Much research has been done and published in recent years on this subject, but it appears that the available programs have not been sufficiently tried and approved to become fully established as indispensable design tools. One area not too well covered at the present time is that of optimal design for moving loads (multiple loading conditions).

Also, most present programs tend to cover too many types of structures and because of this generally become cumbersome to use and appear unfamiliar to the practical designer with not too much background in the more recent developments in structural design.

There are many structural analysis programs in existence, at present. Some are private property of companies or individuals, and some are common knowledge published in books, papers or theses. It is the belief of the author that a design and optimization part should be introduced to most of these structural analysis programs, thus converting them into computer-aided structural design programs. With this part added, they should be of considerably more help to the structural designer than in their present format.

The purpose of this study is to demonstrate a method of conversion to a plane truss analysis program.

CHAPTER 2

THEORETICAL CONSIDERATIONS OF THE PROBLEM

CHAPTER 2

THEORETICAL CONSIDERATIONS OF THE PROBLEM

The analysis program chosen for this study is based on "Direct Element Methods of Truss Analysis", as presented by C.K. Wang [1]. It is a matrix method using a displacement or stiffness approach and programmed for a computer.

The final design and optimization part will be done in two stages: stress ratio iteration at the beginning to bring the design point reasonably close to the optimum. Then in the final stage, linear programing and Simplex method will be used. The nonlinear problems will be linearized by Taylor's expansion taking only the linear terms. In order to establish the critical loading conditions for moving loads, the familiar concept of influence lines will be used..

The material and geometry of the truss will be assumed as given except for the depth of truss where optimal depth will be established in preliminary design. Stress and deflection constraints will be introduced and buckling stresses checked for the chosen discrete sections. Members of the truss will be placed in groups with the same area in order to reduce the number of variables and also for practical design considerations. Also a minimum area will be established to avoid excessive cost of using too thin members. In the final design, limits on max. change per cycle will be set, as the optimization process is based on approximate constraint relationships developed by Taylor's expansion which are more accurate in the ~~close~~ neighborhood of the original design point.

The problem is simple enough as long as the structure does not have many members. Complications arise when the structure has many members as it normally has in a practical case. The matrix methods referred to above can be used efficiently when computers are available. Matrix inversion is always involved and in a practical problem can soon exhaust the available computer memory so that recycling may become uneconomical. Although the computer's capacity is increasing, computer time is still expensive and therefore simplifications and approximations are used in order to keep problems manageable within the available computer capabilities and within the economical limits. Matrix inversions by parts reduces the memory requirements and approximations can be used for finding the influence a change in some variable has on the others, thus, avoiding a full cycle of re-analysis of the structure. In some cases, it might be advantageous to solve the dual instead of the primal linear programming problem.

2.1 PRELIMINARY DESIGN

Before the final design can be started there is a preliminary part to be completed first:

1) Establish the geometry of truss: length of spans, number and type of supports, number of panels and the optimal depth of truss.

2) Dead load: the usual convention of truss design is assumed; all loads are applied at joints only - no bending in members considered. Deck, stringers, cross-beams, ties and bracing are designed and weights for each joint are calculated. These loads remain constant throughout the design process as the panel arrangement is not altered.

Dead load of truss itself for the first analysis cycle is estimated from empirical formulas, for example:

$$W_L = \frac{3}{4}(1 + 0.1\ell) \text{ for roof trusses}$$

W_L = weight of truss (lb per sq.ft of horizontal area supported)

L = span of truss in feet

This load is added to the joint loads computed before.

Weight of truss is recalculated after the first cycle of the design is completed and dead loads on the joints are corrected accordingly.

3) Live load and loading conditions: the analysis program does not perform any additions to get the largest or critical member forces. Therefore, the live load should be located in the critical positions for the various members and added to the corresponding joint loads due to D.L. Analysis is done for these loading conditions and critical member forces chosen for the design of the members. To choose the appropriate LL-positions, influence lines are used which are available after the first cycle of analysis. The value of the influence line coordinate varies as the member areas or truss depths change but the shape will remain the same. As the loads are always at the joints the same influence lines and therefore the same critical loading conditions can be used throughout the whole design process.

4) Member areas for preliminary design: as we are designing a statically indeterminate truss it is important to choose the areas of members for the first cycle as correctly as possible to speed up the convergence. Preliminary design will be an attempted fully stressed design as the member areas

should be in the right ratios to each other which would make member forces correct. The following areas will be used: Let A = area of lower chord.

Then:

Upper chord	= 1.25 A	assumed ratios for starting preliminary design
Lower chord	= 1.0 A	
Diagonals	= 0.5 A	
Verticals	= 0.75 A	

5) Direct element method of truss analysis extended with a list of available sections and a routine for choosing members from the list according to C.S.A. S16 is used for the preliminary design. This program is named DEMTDP. It will reanalyze a truss and redesign it any predetermined number of cycles, checking the total weight of the truss for every cycle. The design will be considered complete for this one depth as soon as the difference between the weight of two consecutive cycles is less than 1% or when the maximum number of cycles is completed. Then the depth of the truss is reduced and the design is repeated for the new depth using as start the same areas as for the preceding design. At the present the program would do five complete designs of the truss for varying depth.

6) Optimal depth of the truss: from these five preliminary designs the optimal depth is chosen considering also the cost of wall and columns for roof trusses or the cost of piers and approaches for bridge trusses. The relationship between the depth of the truss and the weight or cost is displayed on a graph and optimal depth chosen by inspection. This optimal depth with the member areas from the nearest preliminary design cycle is used to start the final design.

7) Constraints in preliminary design: the preliminary design routine has the same constraints as the final design:

Stress - tension and buckling

Deflections

Minimum member area

Identical member areas

2.2 FINAL DESIGN

For the final design the same direct element method of truss analysis with extensions as for the preliminary design is used. The computer program here is called DEMTD and is the main program with 3-subroutines.

It takes as input the optimal depth of the truss as established from the graph with data from the preliminary design. The geometry is taken as in the preliminary design, member areas as in the preliminary design with the depth nearest to the optimal.

This program re-analyses and re-designs the truss for the optimal depth the same way as the trial trusses in the preliminary design. It also checks if deformation constraints are violated and if they are all member areas are increased in the same ratio. This data is taken by the first subroutine MINTW which computes the coefficients for stress and deformation inequalities. These inequalities are compiled, slack variables are added thus turning them into equations suitable for handling in a linear programming problem. Equations are added for the upper and lower bounds on the changes in design variables - member areas in this case, also for minimum areas and identical member areas. Also the objective function for use in the Simplex tableau is compiled.

This data is taken over by a second subroutine LINPROG which adds an infeasibility function and solves the linear programming problem by the Simplex method optimizing the member areas.

The optimized areas from LINPROG are taken over by a third subroutine DEMTR. The structure is analyzed using these areas and they are upgraded if either stress or deformation constraints are violated. Analysis is repeated with these upgraded areas and new members chosen from the list. Then the weight of the truss is computed and compared with the weight in the preceeding cycle. If the difference is less than 0.5%, this is taken as the optimum design. If the difference is more than 0.5% the program returns to MINTW and with new areas the cycle is repeated again until the difference between the weights in the two consecutive cycles is less than 0.5% or the given maximum number of cycles are completed. This last re-design is then accepted as the "optimal design" and the weight of the truss and the member sizes are printed out.

This procedure is composed of existing programs and subroutines with modifications. Some of them will be described in detail in the next sections.

CHAPTER 3

DETAILS OF THE PROPOSED PROCEDURE OF ANALYSIS AND DESIGN

CHAPTER 3

DETAILS OF THE PROPOSED PROCEDURE OF ANALYSIS AND DESIGN

The method used in this study is based on matrix methods for structural analysis combined with linear programming. All these procedures are known and are described in various sources although they are not quite in common use as design office routines yet. Therefore, for the sake of making this study self-contained, the following will be briefly reviewed:

1) The displacement or stiffness method of structural analysis.

2) Direct element method of truss analysis.

3) Influence line ordinates.

4) Stress ratio iteration method of optimum design.

5) Optimum depth of truss.

6) Linear programming and Simplex method.

7) Structural design by linear programming: objective function and constraints for a truss.

8) Coefficients in minimum truss weight equations.

3.1 THE DISPLACEMENT OR STIFFNESS METHOD OF STRUCTURAL ANALYSIS

The displacement method in this study is confined to truss analysis but it is applicable also to almost any type of structural framework. Basically, it is a virtual work method with displacements as unknowns. Results are obtained by solving a number of linear equations, which is done by manipulating the matrix of their coefficients. Hence, the name "Matrix Method". Matrix operations are usually done by computer and availability of digital electronic computer is the main reason why matrix methods are used in structural analysis, nowadays.

To start the analysis, the following matrices are required:

1) $[A]_{NP \times NF}$ - Statics Matrix which expresses for each joint the statical equilibrium between external and internal forces and represents the geometry of the framework.

2) $[S]_{NF \times NF}$ - Member Stiffness Matrix which expresses internal forces in terms of internal deformations. [S] is a diagonal matrix with non-zero members on the main diagonal only.

3) $[P]_{NP \times NLC}$ - External Force Matrix.
From these matrices all the others required for solving the structural analysis problem are developed.

4) $[B]_{NF \times NP} = [A]^T_{NF \times NP}$ - Compatibility Matrix which expresses for each joint the relationship between internal and external deformations.

5) $[C]_{NF \times NLC}$ - Internal Deformation Matrix.

6) $[K]_{NP \times NP} = [ASA^T]_{NP \times NP}$ - External Stiffness Matrix which expresses the external joint forces in terms of joint displacements. Its inverse expresses joint displacements in terms of joint forces.

7) $[X]_{NP \times NLC}$ - External Deformation Matrix.
Part of the solution of the structural analysis problems.

8) $[F]_{NF \times NLC}$ - Internal Force Matrix.
Part of the Solution of the structural analysis problems.

The notations used here are[1]:

NP - Number of degrees of freedom - or number of possible joint forces, or number of possible joint displacements.

---

[1] See also p. "Notations".

NF - Number of members - also number of internal forces "F" or number of member elongations "e".

NLC - Number of loading conditions.

A structural system is analyzed by the following matrix operations:

$$[P]_{NP \times NLC} = [A]_{NP \times NF} [F]_{NF \times NLC} \quad (3.1)$$

(equilibrium equation)

For a statically determinate structure NP = NF, i.e. number of degrees of freedom equals number of internal force. The solution for internal forces of the system is accomplished by solving this first set of equilibrium equations directly as [A] is a square non-singular matrix and can be inverted:

$$[F]_{NF \times NLC} = [A]^{-1}_{NP \times NP} [P]_{NP \times NLC} \quad (3.2)$$

For statically indeterminate structures NF > NP therefore such systems cannot be solved by equilibrium equations alone. Stress-strain and compatibility relationships are also utilized.

$$[F]_{NF \times NLC} = [S]_{NF \times NF} [e]_{NF \times NLC} \quad (3.3)$$

(stress-strain equation)

$$[e]_{NF \times NLC} = [A]^T_{NF \times NP} [X]_{NP \times NLC} \quad (3.4)$$

(compatibility equation)

From Eq. (3.3) and (3.4)

$$[F]_{NF \times NLC} = [SA^T]_{NF \times NP} [X]_{NP \times NLC} \quad (3.5)$$

From Eq. (3.1) and (3.5)

$$[P]_{NP \times NLC} = [ASA^T]_{NP \times NP} [X]_{NP \times NLC} \quad (3.6)$$

Eq. (3.6) is solved for [X]

$$[X]_{NP \times LC} = [ASA^T]^{-1}_{NP \times NP} [P]_{NP \times NLC} \quad (3.7)$$

From Eq. (3.5) and (3.7)

$$[F]_{NF \times NLC} = [SA^T]_{NF \times NP} [ASA^T]^{-1}_{NP \times NP} [P]_{NP \times NLC} \quad (3.8)$$

Let

$$[K]_{NP \times NP} = [ASA^T]_{NP \times NP} \quad (3.9)$$

Then

$$[P]_{NP \times NLC} = [K]_{NP \times P} [X]_{NP \times NLC} \quad (3.10)$$

$$[X]_{NP \times NLC} = [K]^{-1}_{NP \times NLC} [P]_{NP \times NLC} \quad (3.11)$$

$$[F]_{NF \times NLC} = [SA^T]_{NF \times NP} [K]^{-1}_{NP \times NP} [P]_{NP \times NLC} \quad (3.12)$$

3.2 DIRECT ELEMENT METHOD OF TRUSS ANALYSIS

In the displacement method of structural analysis the following matrices are compiled and stored in computer memory:

Statics matrix $[A]_{NP \times NF}$

Compatibility matrix $[A^T]_{NF \times NP}$

Stiffness matrix $[ASA^T]_{NP \times NP}$

For a large structure with many members and degrees of freedom these are large arrays and take up much space in computer memory which may not be available on a smaller machine and would be expensive to occupy, in any case.

In the direct element method, neither statics matrix nor its transpose for the whole structure is ever compiled. Instead of this, the stiffness matrix of the structure is built up directly from the contributions of each member's stiffness which for a truss member is a 4 x 4 matrix only.

$$[AE \quad SE \quad AE^T]_{4 \times 4}$$

$[AE]_{4 \times 1}$ is the statics matrix of one member.

$[SE]_{1 \times 1}$ is the stiffness matrix of one member.

Members are handled one at a time and the matrices for handling are small. When calculating the member forces from known X, the SA^T also is computed for each member separately without having an SA^T matrix for the whole structure in memory.

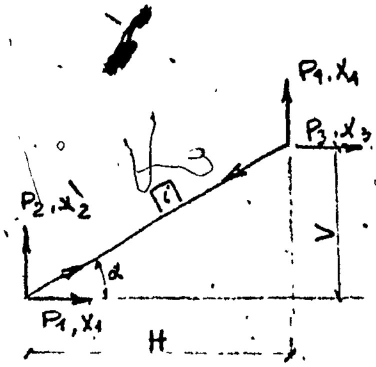
The lengths of the members and direction cosines and sines are also calculated by computer from the given information about each member. Coordinates for each member are given from the initial point as origin: H and V are quantities with plus or minus signs. The positions of the members in the structure are established by the numbering of the degrees of freedom.

<u>Example:</u> <u>Typical Truss Member "i"</u>. Given: H, V from initial point, A area,

NP1,NP2,NP3,NP4 - degrees of freedom numbers in global system.

Computer calculates

$$L_i = \sqrt{H^2+V^2}$$



External forces and displacements

$$S_{Ei} = \frac{E\ A_i}{L_i} \text{ stiffness of member}$$

$$\sin\alpha = \frac{V}{L} = CX$$

$$\cos\alpha = \frac{H}{L} = CY$$

For sign convention, it is assumed that force in a truss member is positive when in tension and displacement x is positive when in the same direction as force P.

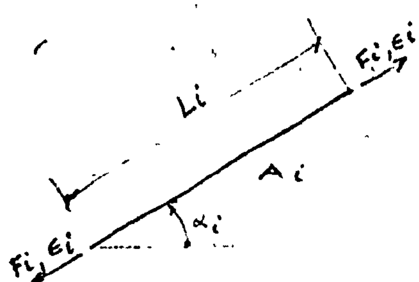
<u>Equilibrium Equations</u>

$$P_1 + Fi\ \cos\alpha = 0 \qquad P_1 = -\ Fi\ \cos\alpha$$

$$P_2 + Fi\ \sin\alpha = 0 \qquad P_2 = -\ Fi\ \sin\alpha$$

$$P_3 - Fi\ \cos\alpha = 0 \qquad P_3 = +\ Fi\ \cos\alpha$$

$$P_4 - Fi\ \sin\alpha = 0 \qquad P_4 = +\ Fi\ \sin\alpha$$



Internal forces and deformations

<u>Compatibility Equation</u>

$$\varepsilon_i = -\ X_1\ \cos\alpha - X_2\ \sin\alpha + X_3\ \cos\alpha + \\ + X_4\ \sin\alpha$$

In matrix notation:

<u>Equilibrium Equation</u>

$$\begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{Bmatrix} = \begin{Bmatrix} -\cos \\ -\sin \\ +\cos \\ +\sin \end{Bmatrix} [Fi] \quad \text{or} \quad [P]_{4\ x\ 1} = [Ai]_{4\ x\ 1} \times [E]_{1\ x\ 1}$$

Compatibility Equation

$$\varepsilon i = [-\cos\alpha \; - \sin\alpha \; + \cos\alpha \; + \sin\alpha] \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{Bmatrix}$$

or $$\varepsilon i = [AEi]^T_{1 \times 4} \; [X]_{4 \times 1}$$

$$[AEi] = \begin{bmatrix} -\cos \\ -\sin \\ +\cos \\ +\sin \end{bmatrix}$$ statics matrix of a member

$$[AEi]^T = [-\cos\alpha \; - \sin\alpha \; + \cos\alpha \; + \sin\alpha]$$

Compatibility matrix of a member

External stiffness matrix of a truss member:

$$[KE]_{4 \times 4} = [AE \times SE \times AE^T] = \begin{bmatrix} -\cos\alpha \\ -\sin\alpha \\ +\cos\alpha \\ +\sin\alpha \end{bmatrix} \times \frac{E\varepsilon \; Ai}{Li} \times$$

$$[-\cos\alpha \; - \sin\alpha \; + \cos\alpha \; + \sin\alpha]$$

$$[KE]_{4 \times 4} = \frac{E \; Ai}{Li} \begin{bmatrix} \cos^2\alpha & \cos\alpha \sin\alpha & -\cos^2\alpha & -\cos\alpha \sin\alpha \\ \sin\alpha \cos\alpha & \sin\alpha^2 & -\sin\alpha \cos\alpha & -\sin^2\alpha \\ -\cos\alpha^2 & -\cos\alpha \sin\alpha & \cos^2\alpha & \cos\alpha \sin\alpha \\ -\sin\alpha \cos\alpha & -\sin^2\alpha & \sin\alpha \cos\alpha & \sin^2\alpha \end{bmatrix}$$

$$[Pi]_{4 \times 1} = [KE]_{4 \times 4} \times \{X\}_{4 \times 1}$$

Such external stiffness matrices are developed for each member of the truss and from these elements the external stiffness matrix for the structure is built by adding all the stiffnesses of individual members in each joint:

$$[K]_{NP \times NP} = \sum_{i=1}^{NM} [KEi]_{4 \times 4}$$ - external stiffness matrix of the structure

With [K] available, joint deformations X and member forces F are computed from Eqs. (3.11) and (3.12).

$$[X]_{NP \times NLC} = [K]^{-1}_{NP \times NP} \times [P]_{NP \times NLC}$$

$$[F]_{NF \times NLC} = [SA^T]_{NF \times NP} [K]^{-1}_{NP \times NP} [P]_{NP \times NLC}$$

3.3 INFLUENCE LINES

By definition the influence line of a member force or a joint deflection consists of the trace of ordinates representing to some scale, the value of that force or deflection caused by a unit load in the position of the ordinate.

Ordinates of influence lines can be computed from the equations (3.11) and (3.12) by replacing the external force matrix $[P]_{NP \times NLC}$ with a unit matrix $[I]_{NP \times NP}$. This is equivalent to introducing NP new loading conditions with one unit load at a time acting in the direction of each degree of freedom:

$$[Fi]_{NF \times NP} = [SA^T]_{NF \times NP} [K]^{-1}_{NP \times NP} \cdot [I]_{NP \times NP} =$$

$$= [SA^T]_{NF \times NP} \cdot [X]_{nP \times NP}$$

$$[X]_{NP \times NP} = [K]^{-1}_{NP \times NP} \cdot [I]_{NP \times NP}$$

With the direct element method of truss analysis, an inverted global stiffness matrix and therefore ordinates for influence lines of deflections are available after the first cycle of analysis is completed.

Matrix $[SA^T]$ is not available for the structure but an external force matrix can be extended for the first cycle of analysis with a unit matrix and influence line ordinates printed out with the member force matrix.

For simple structures, critical loading conditions can be established by inspection of the geometry. With more complicated ones, as statically indeterminate trusses with internal indeterminacy and continuous over several spans this can be best done by inspection of influence lines. It is quite important for web members and also for chords since forces in chords can also change sign due to the locations of the loads.

3.4 STRESS RATIO ITERATION METHOD OF OPTIMUM DESIGN

This method attempts to achieve a fully stressed design assuming this to be the same as the optimum design. It has been proven by various authors , that fully stressed design is not always the optimum. However, as this method will be used for preliminary design only, in order to establish the optimum depth of the truss, it is useful for improving the original design point so that the more elaborate (linear programming) methods can be profitably applied in the final design. Analysis with the assumed member areas will give member forces for all loading conditions :

$$[F] = [SA^T][ASA^T]^{-1}[P] \qquad (3.13)$$

By introducing the diagonal area matrix $[\underline{A}]_{NF \times NF}$ stresses in members will be given by the following equations:

$$\begin{aligned}[\sigma]_{NF \times NLC} &= [\underline{A}]^{-1}\,[F]_{NF \times NLC} \\ &= [\underline{A}]^{-1}[SA^T][ASA^T]^{-1}\,[P] \\ &= [\underline{A}]^{-1}_{NF \times NF}[SA^T]_{NF \times NP}[X]_{NP \times NLC} \qquad (3.14)\end{aligned}$$

Let $(A_j)^o$ $(j = 1,2,3 \ldots NF)$ = member areas of the previous cycle.

$(\dot{\sigma}_{jr}\dot{z})^{o}$ $(j = 1,2,3 \ldots .\ NF,\ r = 1,2 \ldots .\ NLC)$=stress in member j at the loading condition r.

Then a new, improved set of member areas (new design point) is computed as follows:

$$(A_j)^1 = (A^o_j)[\max [\frac{(\sigma_j, r)^o}{\sigma_{j,r}\text{allowable}},\ r = 1,2 \ldots NLC]] \quad (3.15)$$

This method would give fully stressed or optimal member areas for a statically determinate truss in one cycle.

For a statically indeterminate truss or one subject to deflection constraints, this method will produce neither a fully stressed nor an optimal design but in a few cycles it will lead to an improved design point. Objective function will be the minimum weight of the truss.

$$\text{Minimize} \quad Z = \sum_{j=1}^{NM} L_j . A_j . \rho \qquad (3.16)$$

The design will be considered optimal if the weight of the truss does not differ from the previous cycle by more than 1%. The max. number of cycles will be limited to 10.

3.5 OPTIMAL DEPTH OF A TRUSS

Our criterion for optimality was the weight of the truss. The reasoning that the lowest weight means the lowest cost is quite correct if the depth of the truss is given as a design condition. But in some cases, the depth of the truss is left for a structural designer to choose. If so, his criterion again should be the minimum cost, except that here, other items besides the cost of the truss itself would have to be considered, as they are affected by the dimensions of the truss.

Taking the simple case of a parallel chord roof truss of an industrial building, the depth of the truss will influence the area of the exterior wall and if columns are used as end posts for trusses, the length and cost of the columns will also change.

If we have a bridge truss with deck on top of the trusses and specific clearance below, the height of approaches will be affected and the cost of it would have to be considered, together with the cost of the truss, itself.

Optimization of geometry has been treated by various authors, but it is not a major subject of this study. The proposed method here, is to produce several preliminary designs of the truss with varying depths, calculate the cost of the truss, together with other items involved, and display the cost versus the depth in a graph. It appears reasonable to assume that the change in the cost between the various depths of the truss will be gradual, and therefore, the optimal depth will be chosen by inspection.

For a roof truss, the comparative cost will be expressed as follows.

$$C = (W_t . C_t) + 2(b.h.cw) + (h.W_{col}.n.C_{col}) \quad (3.17)$$

where

W_t = weight of truss

C_t = cost of truss $/lbs.

b = spacing of trusses ft.

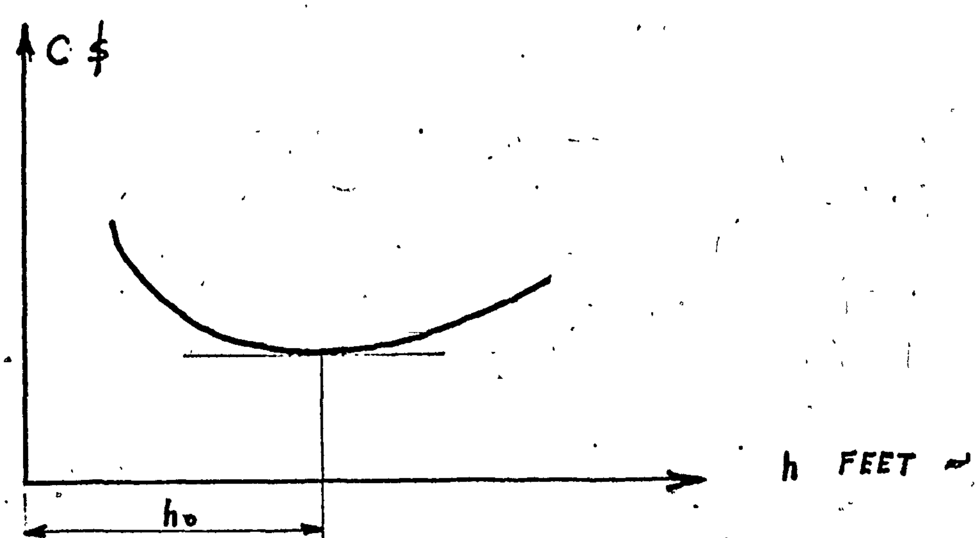
cw = cost of wall $/S.F.

W_{col} = weight of one supporting column lbs/L.F.

n = number of supporting columns

C_{col} = cost of column $/lbs

h = depth of truss ft.



ho OPTIMAL DEPTH OF TRUSS.

It is not always that the designer can choose the depth of the truss, but if he can, it becomes a very important part of the overall economy and most of the time, more real savings can be made by choosing the correct depth than by the best choice of member sizes for a poorly chosen depth.

3.6 LINEAR PROGRAMMING AND THE SIMPLEX METHOD

Mathematical programming of which linear programming is a part, solves problems of maximization or minimization of a numerical function with a number of variables. The classical method is to use differential calculus if it is an unconstrained optimization problem. If the values of solution variables are restricted by constrain[illegible]losed solution in some cases can be obtained by the use of L[illegible]ian multipliers. This is constrained optimization.

Besides the problems with closed solutions there are others where optimization can only be achieved by iterative methods. These are the mathematical programming problems defined as "the determination of the optimum allocation of limited resources to meet given objectives"[4]. Mathematically, these are problems of constrained optimization: optimize an objective function subject to a number of constraints.

Mathematical programming problems are usually divided in two large groups depending on the nature of objective functions and constraints. If the objective function and constraints are linear, we have a linear programming problem. If either the objective function or the constraints or both objective function and constraints are nonlinear, it is a nonlinear programming problem.

Although most practical problems would fall into nonlinear programming group, there is a great advantage in reducing them to linear programming problems, as these are much easier to solve. The Simplex method for solving linear programming problems is a very well developed and efficient algorithm. It is suitable for hand operation in tableau form and it can also be programmed for a digital computer.

The structural optimization or minimum weight design belongs rightly with nonlinear programming problems. In this study - truss design - objective function is linear but stress and deflection constraints are nonlinear. To make the solution practical, the constraints are linearized resulting in the loss of some accuracy but gaining very much in computational efficiency.

The Simplex Method:

The present study uses linear programming and the Simplex method. As such these will be briefly reviewed here. The theory of linear programming and the Simplex method are well documented in many standard texts. Therefore, only the computational aspects concerning this particular problem will be reviewed.

All linear programming problems can be expressed in standard form.

$$\left.\begin{array}{l} a_{11}\, x_1 + a_{12}\, x_2 + \;\ldots\ldots\; a_{1n}\, x_n = b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \;\ldots\ldots\; a_{2n}\, x_n = b2 \\ \vdots \\ a_{m1}\, x_1 + a_{m2}\, x_2 + \;\ldots\ldots\; a_{mn}\, x_n = b_m \\ x_j \geq o \qquad j = 1,n \end{array}\right\} \text{constraint equations} \tag{3.18}$$

$$c_1\, x_1 + c_2\, x_2 + \;\ldots\ldots\; c_n\, x_n = \min \text{ objective function} \tag{3.19}$$

The various constants in linear programming problems are:

C_j = cost coefficients, (objective function)

a_{ij} = structural coefficients (constraints)

b_i = stipulations, (constraints)

The optimization in a linear programming problem can either be minimization or maximization. If the object is maximization, minimization problems may be formed by multiplying the objective function by -1.

The actual constraint conditions in general, may include inequalities of both "greater than" and "less than" signs or a mixture of inequalities and equations. In order to replace inequality constraint conditions by equations, the concept of slack variables is used. A constraint condition of "less than" sign is turned into an equation by adding a slack variable.

$$a_{i1} x_1 + a_{i2} x_2 + \ldots\ldots a_{iq} x_q \leq b_i$$

is replaced by

$$a_{i1} x_1 + a_{i2} x_2 + \ldots\ldots a_{iq} x_q + x_{q+1} = b_i$$

A constraint condition of "greater than" sign is turned into an equation by subtracting a slack variable.

$$a_{i1} x_1 + a_{i2} x_2 + \ldots\ldots a_{iq} x_q \geq b_i$$

is replaced by

$$a_{i1} x_1 + a_{i2} x_2 + \ldots\ldots a_{iq} x_q - x_{q+1} = b_i$$

A slack variable, in both cases, is a non-negative variable.

Linear programming problems can also be expressed in inequality form with all constraints having the same inequality sign. In a minimization problem, we use the "greater than" inequality for constraints, as it is also consistent with the objective function which can be taught as having a "greater than" sign.

If there is a constraint inequality of the "less than" sign, it can be converted into the "greater than" sign by multiplying both sides by -1.0. If there is an equation, it can be replaced by inequalities.

$$a_{i1}\ x_1 + a_{i2}\ x_2 + \ldots\ldots + a_{iq}\ x_q = b_i$$

replaced by

$$\begin{cases} a_{i1}\ x_1 + a_{i2}\ x_2 + \ldots\ldots + a_{iq}\ x_q \geq b_i \\ \\ a_{i1}\ x_i + a_{i2}\ x_2 + \ldots\ldots + a_{iq}\ x_q \leq b_i \end{cases}$$

In minimization problems, we use "less than" inequalities, again to be consistent with the objective function.

Both the minimization and maximization problems in inequality forms can be transformed into standard forms by adding or subtracting non-negative slack variables.

If the number of constraint equations is m and number of the variables is n, there can be three cases to consider

1) $m = n$: there is one unique solution to the problem which is also optimal.

2) $m > n$: there can be no solution unless $m - n$ equations are linearly dependent.

3) $m < n$: this is the case that we are practically interested in.

The solution for any set of linear equations can be only found when the number of equations is equal to the number of variables. Linear programming solves this problem by systematically selecting sets of m variables from the available "n" variables, solving those sets and comparing the results of the objective function. The remainder of the n-m variables is assumed to have the value of zero. The "m" variables are called basic variables, the n-m variables are called non-basic variables. The set of values for the basic variables are called basic solutions. If the basic solutions also satisfies the non-negativity requirement,it is called a basic feasible solution. Instead of investigating all possible basic solutions, linear programming is only concerned with the basic feasible solutions,among which the optimal solution is chosen. This solution process can be graphically demonstrated if there are no more than two variables. It is hardly ever the case in structural problems and therefore is not demonstrated here. The procedure suitable for handling large numbers of variables is the Simplex method introduced by Dantzig. [5]

It is an iterative procedure for finding the optimal solution. Starting with the standard form which already contains the necessary slack variables, "m" variables are chosen as the basic and system solved, reducing it to canonical form. Values of the basic variables are put into the objective function which then does contain non-basic variables only. Then the objective function is examined: if there is any possibility of decreasing its value by introducing into the basis one of the non-basic variables and removing a basic one. The canonical form of constraints and the corresponding objective function looks like this:[3]

$$
\begin{array}{lllllll}
x_1 & + \bar{a}_{1,m+1} X_{m+1} & + \dots + \bar{a}_{1,s} X_s & + \dots + \bar{a}_{1,n} X_n & = \bar{b}_1 & \\
\quad x_2 & + \bar{a}_{2,m+1} X_{m+1} & + \dots + a_{2,s} X_s & + \dots + \bar{a}_{2,n} X_n & = \bar{b}_2 & \\
 & \vdots & & \vdots & \vdots & \\
\qquad x_r & + \bar{a}_{r,m+1} X_{m+1} & + \dots + \bar{a}_{r,s} X_s & + \dots + \bar{a}_{r,n} X_n & = \bar{b}_r & 3.20 \\
\qquad\quad x_m & + \bar{a}_{m,m+1} X_{m+1} & + \dots + \bar{a}_{m,s} X_s & + \dots + a_{m,n} X_n & = \bar{b}_m & \\
 & + \bar{c}_{m,m+1} X_{m+1} & + \dots + \bar{c}_s X_s & + \dots + \bar{c}_n X_n & = & \\
 & & & = \min Z - \bar{Z} & &
\end{array}
$$

This is one solution where $x_i = \bar{b}_i$, $i = 1,m$ and min $Z = \bar{Z}$, $\bar{Z} = \sum_{i=1}^{m} c_i \bar{b}_i$. If all $\bar{b}_i \geq 0$, $i = 1, m$, this is also a feasible solution. If all coefficients $\bar{c}$ in the objective function are positive, this is an optimal solution because giving a positive value to any of the non-basic variables would increase the min Z. But, if some coefficients $\bar{c}$ are negative, then by giving a positive value to the corresponding non-basic variable, min Z can be reduced. We choose the non-basic variable with the negative $\bar{c}_s$ of largest absolute value. Next, we have to select which basic variable is to be deleted so that we gain the largest reduction of Min Z, at the same time not violating the non-negativity requirement for the other basic variables. This is done by examining the coefficients in the columns "s" and "$\bar{b}$". When X_s is increased from 0 to a positive quantity, there will be one X_r which will tend to become equal to zero first. Further increase of x_s would make it negative, thus giving infeasible solution. So we compare ratios $\frac{\bar{b}_i}{\bar{a}_{is}}$ for $i = 1,m$ and choose the smallest, say $\frac{\bar{b}_r}{\bar{a}_{rs}} = X_s$. This is the limiting value for X_s and therefore x_r is removed from the basis. With $\bar{a}_{rs}$ as pivot, the basis again is solved, substituted into objective function and cycle repeated until all coefficients in objective function become positive or zero. This is then the optional solution.

If any of $\bar{a}_{is}$ are negative, comparison is done among the positive ones only. If all $\bar{a}_{is}$ are negative no upper limit exists for X_s and solution is unbounded - no optimal solution exists.

This is basically the Simplex method without mentioning any of the mathematical complications which can arise. The solution to maximization problem is identical to that of the minimization except that the largest positive coefficient $\bar{c}_s$ is chosen as we are interested in increasing the objective function as much as possible.

There is a device to find a basic feasible solution in order to start the problem. It will also establish if a feasible solution exists and will end the problem if feasible solution does not exist. This is done by extending the problem and introducing a set of non-negative "artificial" variables x_i, $i = n+1, n+m$ one for each constraint equation and making all coefficients in the "b" column positive - multiplying the equations by -1, if required. The extended set of constraint equations will be as follows.

$$\begin{array}{llllllll} a_{11} x_1 + a_{12} x_2 + & \dots & a_{1n} x_n + x_{n+1} & & & = b_1 \\ a_{21} x_1 + a_{22} x_2 + & \dots & a_{2n} x_n & + x_{n+2} & & = b_2 \\ \vdots & & & & \ddots & \\ a_{m1} x_1 + a_{m2} x_2 + & \dots & a_{mn} x_n & & + x_{n+m} & = b_m \end{array}$$

Let $x_{n+1} + x_{n+2} + \dots + x_{n+m} = w$, the sum of artificial variables, which is called infeasibility function. If min $W = 0$, a feasible solution exists for the linear programming problem. If min $W \neq 0$, there is no feasible solution. The infeasibility function is modified the same way as the objective function in the Simplex method: the values for artificial variables are substituted in infeasibility functions. Then the same testing and manipulation is applied as before until we find that $w = 0$, or that $w \neq 0$. If $w = 0$, we know that the feasible solution exists and proceed to solve the min Z of the objective function, starting with the last basis from the infeasibility function. The linear programming problem in this study is solved in this way.

Duality:

There is another important feature of linear programming problems which has some use for structural problems with large numbers of constraints. It is called duality.

Given here without proof, it is as follows.

If there is a minimization problem which has a feasible solution, then there is also a related maximization problem which has the same optimum value. By utilizing this rule, it is sometimes possible to reduce the basis matrix and simplify finding the solution. If the minimization problem is called "primal" - the maximization is "dual". The solution variables of primal problem are found in the dual tableau under the value of slack variables and vice-versa.

To show this relationship, the linear programming problem is stated in inequality form.

$$x_j \geq 0, \quad j = 1,q$$

$$\begin{array}{llllll} a_{11}\, x_1 + a_{12}\, x_2 + & \ldots & a_{1q}\, x_q & \geq b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + & \ldots & a_{2q}\, x_q & \geq b_2 \\ a_{p1}\, x_1 + a_{p2}\, x_2 + & \ldots & a_{pq}\, x_q & \geq b_p \\ c_1\, x_1 + c_2\, x_2 + & \ldots & c_q\, x_q & = \min Z \end{array}$$

We call this problem primal and the set of variables $xj \geq 0$, $j = 1,q$ primal variables.

The corresponding dual problem with its set of dual variables $y_i \geq 0$, $i = 1,p$ has an inequality form as follows.

$$y_i \geq 0, \quad i = 1,p$$

$$a_{11}\, y_1 + a_{21}\, y_2 + \quad \cdots \quad a_{p1}\, y_p \leq c_1$$

$$a_{12}\, y_1 + a_{22}\, y_2 + \quad \cdots \quad a_{p2}\, y_p \leq c_2$$

$$a_{1q}\, y_1 + a_{2q}\, y_2 + \quad \cdots \quad a_{pq}\, y_p \leq c_q$$

$$b_1\, y_1 + b_2\, y_2 + \quad \cdots \quad b_p\, y_p = \text{Max } A$$

By definition:

1) The matrix of coefficients of constraint inequalities for dual is a transpose of that for primal problem.

2) The cost coefficients "c" of primal problem are stipulants in the "b" column in dual.

3) The "b" column in dual is the vector of transpose of cost coefficients "c".

4) The inequality sign of primal is reversed for dual

5) Minimization in primal problem is changed to maximization in dual.

6) The optimal values for both primal and dual problem are identical.

<u>Sensitivity Analysis:</u>

To solve a linear programming problem, is generally a rather lengthy and cumbersome process. Besides, if it is a real life problem, none of the coefficients are exact and are subject to change as the conditions change. Therefore, in some linear programming applications, it is quite important to know without reworking the whole problem what the effects on the solution will be if any of the coefficients are changed.

This is called sensitivity analysis. It has limited application in this particular study as the process is automated: a new linear programming problem is solved every time when the coefficients are changed - as areas are changed after each cycle.

3.7 STRUCTURAL DESIGN BY LINEAR PROGRAMMING: OBJECTIVE FUNCTION, CONSTRAINTS

Member forces and joint deformations are non-linear functions of member areas for a statically indeterminate system. The problem in this case is to express change in member forces or joint deformations due to a change in a member's area. Member forces and joint deformation are continuous functions of member areas, therefore derivatives exist, and the change can be expressed in Taylor's series:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + $$

$$+ \frac{h^3}{3!} f'''(x) + \ldots\ldots$$

As a linear approximation only the first two members are used, the rest being second order small:

$$f(x+h) = f(x) + h\, f'(x)$$

Member areas will be used as the only independent variables in this study.

3.7.1 Objective function

For truss problem objective function is linear:

$$\text{Minimize: } W = \sum_{j=1}^{NM} \rho\, L_j . A^o_j . u_j \quad \text{- sum of weights of truss members} \tag{3.21}$$

Where

NM = total number of members

ρ = specific weight of material

L_j = length of member j

A^o_j = initially assumed member area

$$u_j = 1 + \frac{\Delta A_j}{A^o_j} \qquad u_j \geq 0 \tag{3.22}$$

ΔA_j = change in member area

$\{u_j\}_{NM \times 1}$ = solution vector - multipliers for each member area

$$W_o = \sum_{j=1}^{NM} \rho L_j . A^o_j = \text{total weight of initially assumed systems} \tag{3.23}$$

3.7.2 Stress constraints

$$f = + \frac{F}{A} \quad \text{tension stress}$$

$$f = - \frac{F}{A} \quad \text{compression stress}$$

Stress constraints will be formulated in terms of member forces.

$$F = A_o f \tag{3.24}$$

$$\left.\begin{array}{lll} F_i\text{new} \le (F_i\text{new})\text{all} & \text{when} & F^o \ge 0 \\ F_i\text{new} \ge -(F_i\text{new})\text{all} & \text{when} & F_o < 0 \end{array}\right\} \text{stress constraints} \tag{3.25}$$

F^o_i = acting force in the originally assumed member with area A^o_j

$F_i$new = acting force in the originally assumed member with new area $A_j = A^o_j + \Delta A_j$

$$(F_i new)_{all} = f_i all\ (A^o_j + \Delta A_j) \tag{3.26}$$

$f_i all = 0.66\ F_y$ allowable stress (tension)

$f_i all =$ buckling stress (compression)

$$F_i new = F_i^o + \Delta F_i \tag{3.27}$$

This change in member force ΔF_i is associated with the change in the corresponding design parameters and will be approximated by a single term of Taylor's expansion:

$$\Delta F_i = F_i - F_o = \frac{\partial F_i}{\partial A_i} \cdot \Delta A_i \text{ change in force } F_i \text{ due to change in area } A_i$$

But the force F_i will be affected also by the changes in other member areas. Therefore

$$\Delta F_i = \frac{\partial F_i}{\partial A_1} \Delta A_1 + \frac{\partial F_i}{\partial A_2} \cdot \Delta A_2 + \frac{\partial F_i}{\partial A_3} \Delta A_3 + \dots \frac{\partial F_i}{\partial A_i} \cdot \Delta A_i + \dots + \frac{\partial F_i}{\partial A_{NM}} \Delta A_{NM} = \sum_{j=1}^{NM} \frac{\partial F_i}{\partial A_j} \Delta A_j \tag{3.28}$$

$$F_i new = F^o_i + \Delta F_i = F^o_i + \sum_{j=1}^{NM} \frac{\partial F_i}{\partial A_j} \Delta A_j \tag{3.29}$$

$$i = 1, 2 \dots NM$$

From Eqs. (3.25), (3.26) and (3.29)

$$F^o_i + \sum_{j=1}^{NM} \frac{\partial F_i}{\partial A_j} \Delta A_j \quad \begin{cases} \leq (f_i)all(A^o_i + \Delta A_i), & F^o_i \geq 0 \\ \geq (f_i)all(A^o_i + \Delta A_i), & F^o_i < 0 \end{cases}$$

Stress constraints will be generated for each member force in each loading condition:

$$\left[\left[\left[F^o_{ik} + \sum_{j=1}^{NM} \frac{\partial F_{ik}}{\partial A_j} \Delta A_j \quad \begin{matrix} \leq (f_{ik})all(A^o_i + \Delta A_i), & F^o_{ik} \geq 0 \\ \geq (f_{ik})all(A^o_i + \Delta A_i), & F^o_{ik} < 0 \end{matrix} \right] \right.\right.$$

$$i = 1, NF \Big] \; k = 1, \; NLC \Big] \tag{3.30}$$

Solution variable in stress constraint inequality is ΔA = change in member area which may be positive or negative, increase or decrease of area. But solution will be sought by linear programming and Simplex method where variables cannot be negative. Therefore the same solution variable as for the objective function must be introduced. The following reasoning is used :

Let

$$\Delta F_{ijk} = \frac{\partial F_i}{\partial A_1} A^o_1 + \frac{\partial F_i}{\partial A_2} A^o_2 + \frac{\partial F_i}{\partial A_3} A^o_3 + \ldots$$

$$\ldots \frac{\partial F_i}{\partial A_i} A^o_i + \ldots \frac{\partial F_i}{\partial A_{NM}} A^o_{NM} = \sum_{j=1}^{NM} \frac{\partial F_i}{\partial A_j} A^o_j =$$

$= 0$ - change in force "i" due to change in areas

A^o_1 to A^o_{NM} by 100% - all members changed in same ratio, therefore, no change in member forces.

Then

$$F_{ik}new - F^o_{ik} = \frac{\partial F_i}{\partial A_1} . A^o_1 . \frac{\Delta A_1}{A^o_1} + \frac{\partial F_i}{\partial A_2} . A^o_2 . \frac{\Delta A_2}{A^o_2} +$$

$$+ \frac{\partial F_i}{A_3} . A^o_3 . \frac{A_3}{A^o_3} + \ldots + \frac{\partial F_i}{\partial A_{NM}} A^o_{NM} \frac{\Delta A_{NM}}{A^o_{NM}} =$$

$$= \sum_{j=1}^{NM} \Delta F_{i.j.k.} \frac{\Delta A_j}{A^o_j}$$

Stress constraint inequalities now are:

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta F_{ijk} \cdot \frac{\Delta A_j}{A^o_j} + F^o_{ik} \quad \begin{array}{l} \leq (f_{ik})all(A^o_i + \Delta A_i),\ F^o_{ik} \geq 0 \\ \geq (f_{ik})all(A^o_i + \Delta A_i),\ F^o_{ik} \leq 0 \end{array} \right] i = 1, NF \right] k = 1, NLC \right] \quad (3.31)$$

Add $\sum_{j=1}^{NM} \Delta F_{ijk}$ to both sides where

$$\sum_{j=1}^{NM} \Delta F_{i.j.k} = \Delta F_{i1k} + \Delta F_{i2k} + \ldots \Delta F_{iNMk}$$

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta F_{ijk} \cdot \frac{\Delta A_j}{A^o_j} + \sum_{j=1}^{NM} \Delta F_{ijk} + F^o_{ik} \begin{array}{l} \lesseqgtr (f_{ik})all(A^o_i + \Delta A_i) + \\ \geq (f_{ik})all(A^o_i + \Delta A_i) + \\ + \sum_{j=1}^{NM} \Delta F_{ijk} \; ; \; F^o_{ik} \geq 0 \\ + \sum_{j=1}^{NM} \Delta F_{ijk} \; ; \; F^o_{ik} < 0 \end{array}\right] i = 1,NF\right] k = 1,NLC\right] \quad (3.32)$$

But $\sum_{j=1}^{NM} \Delta F_{ijk} \frac{\Delta A_j}{A^o_j} + \sum_{j=1}^{NM} \Delta F_{ijk} = \frac{\partial F_i}{\partial A_1} \cdot A^o_1 \frac{\Delta A_1}{A^o_1} + \frac{\partial F_i}{\partial A_1} : A^o_1 +$

$$+ \frac{\partial F_i}{\partial A_2} A^o_2 \frac{\Delta A_2}{A^o_2} + \frac{\partial F_i}{\partial A_2} \cdot A_2 +$$

$$+ \frac{\partial F_i}{\partial A_3} \cdot A_3 \cdot \frac{\Delta A_3}{A^o_3} + \frac{\partial F_i}{\partial A_3} \cdot A_3 + \ldots\ldots + \frac{\partial F_i}{\partial A_{NM}} \cdot A^o_{NM} \cdot \frac{\Delta A_{NM}}{A^o_{NM}} +$$

$$+ \frac{\partial F_i}{\partial A_{NM}} \cdot A^o_{NM} = \frac{\partial F_i}{\partial A_1} A^o_1 \left(\frac{\Delta A_1}{A^o_1} + 1\right) + \frac{\partial F_i}{\partial A_2} \cdot A^o_2 \left(\frac{\Delta A_2}{A^o_2} + 1\right) +$$

$$+ \frac{\partial F_i}{\partial A_3} A^o_3 \left(\frac{\Delta A_3}{A^o_3} + 1\right) + \ldots\ldots + \frac{\partial F_i}{\partial A_{NM}} A^o_{NM} \left(\frac{\Delta A_{NM}}{A^o_{NM}} + 1\right) =$$

$$= \sum_{j=1}^{NM} \frac{\partial F_i}{\partial A_j} A^o_j \left(\frac{\Delta A_j}{A^o_j} + 1\right) = \sum_{j=1}^{NM} \Delta F_{ijk} \cdot A^o_{jk} \, u_j$$

Stress constraint inequalities become

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta F_{ijk} \cdot u_j + F^o_{ik} \quad \begin{array}{l} \leq (f_{ik})\text{all}.A^o_i \cdot u_i + \sum_{j=1}^{MN} \Delta F_{ijk}; \\ \geq (f_{ik})\text{all}.A^o_i \cdot u_i + \sum_{j=1}^{MN} \Delta F_{ijk}; \end{array}\right.\right.\right.$$

$$\left.\left.\left.\begin{array}{ll} \text{for} & F^o_{ik} \geq 0 \\ \text{for} & F^o_{ik} < 0 \end{array}\right] \; i = 1,NF \right] \; K = 1,MLC \right] \qquad (3.33)$$

But $\sum_{j=1}^{NM} \Delta F_{ijk} = 0$ as it is the sum of all changes in a member force F_i due to 100% increase in all member areas. This is a known theorem that by increasing all member stiffnesses of a structural system in the same ratio the relative stiffness of the system does not change and neither do the member forces change.

Stress constraint inequalities in final form are

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta F_{ijk} \cdot u_j - (f_{ik})\text{all}.A^o_i \cdot u_i \quad \begin{array}{l} \leq - F^o_{ik}; \text{for } F^o_{ik} \geq 0 \\ > - F^o_{ik}; \text{for } F^o_{ik} < 0 \end{array}\right] \; i = 1,NF \right] \; K = 1,NLC \right] \qquad (3.34)$$

where

ΔF_{ijk} = change in member force F_i due to 100% increase of members area A^o_j at loading condition "k".

$$u_j = 1 + \frac{\Delta A_j}{A^o_j} \quad \text{solution variable for area } A^o_j \tag{3.35}$$

A^o_j = original area of member j

ΔA_j = change in area of member j

NF = numbers of critical forces

NM = number of members

$(f_{ik})all$ = allowable stress in member "i" at loading condition "k". This is a signed quantity: Tension = plus, compression = minus

F^o_{ik} = original member force - signed quantity plus or minus.

3.7.3 Deformation constraints

Joint deflections are limited to a maximum X_{all}.

$$\left.\begin{aligned} X_{new} &\leq X_{all}, \quad x^o > 0 \\ &\geq X_{all}, \quad x^o < 0 \end{aligned}\right\} \text{deformation constraints} \tag{3.42}$$

where

X_{new} = new deformation

X_{all} = allowable deformation

X_o = deformation in the initial system

Using the same reasoning as for the stress constraints:

$$(X_i)_{new} = X_i^o + \Delta X_i = X_i^o + \sum_{j=1}^{NM} \frac{\partial X_i}{\partial A_j} \Delta A_j \quad (3.43)$$

Constraint inequality for deformation i:

$$X_i + \sum_{j=1}^{NM} \frac{\partial X_i}{\partial A_j} . \Delta A_j \begin{array}{l} \leq (X_i)_{all},\ x_i^o \geq 0 \\ \geq (X_i)_{all},\ x_i^o < 0 \end{array} \quad (3.44)$$

Constraint inequality for any deformation i under the k loading conditions:

$$\left[\left[\left[X^o{}_{ik} + \sum_{j=1}^{NM} \frac{\partial x_i}{\partial A_j} . \Delta A_j \begin{array}{l} \leq (x_i)_{all}, x^o{}_i \geq 0 \\ > (x_i)_{all}\ x^o{}_i < 0 \end{array} \right] \right. \right.$$

$$\left. i = 1,NP \right] \left. K = 1,NLC \right] \quad (3.45)$$

NP = total number of critical deformations.

For numerical evaluation the same procedure is followed as for stress constraints.

Introducing again Δx_{ijk} = change in deformation i due to 100% increase of the member area j in the loading condition K. Eq. (3.45) becomes

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta x_{ijk} \frac{\Delta A_j}{A^o_j} + X^o_{ik} \quad \begin{matrix} \leq (x_{ik})_{all}; X^o_{ik} > 0 \\ \geq (x_{ik})_{all}; X^o_{ik} < 0 \end{matrix}\right] \quad i = 1,NP\right] \quad K = 1,NLC\right] \tag{3.46}$$

Add $\sum_{j=1}^{NM} \Delta x_{ijk}$ to both sides, introduce variable

$$u_j = 1 + \frac{\Delta A_j}{A_j^o}$$

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta x_{ijk}\, u_j \quad \begin{matrix} \leq (x_{ik})_{all} - X^o_{ik} + \sum_{j=1}^{NM} \Delta x_{ijk}; & X^o_{ik} \geq 0 \\ \geq (x_{ik})_{all} - X^o_{ik} + \sum_{j=1}^{NM} \Delta x_{ijk}; & X^o_{ik} < 0 \end{matrix}\right] \quad i = 1,NP\right] \quad k = 1,NLC\right] \tag{3.47}$$

3.7.4 Numerical Evaluation of Stress and Deformation Constraints

For numerical evaluation of ΔF_{ijk} and ΔX_{ijk} we return to the analysis part of this study which gave the following relationships:

$$P = AF \tag{3.1}$$

$$F = SA^T X \tag{3.5}$$

$$K = ASA^T \tag{3.9}$$

$$X = K^{-1} P \tag{3.11}$$

$$F = SA^T ASA^T P \tag{3.12}$$

The derivatives of forces are required with respect to changes in the design variables.

Let

$\Delta F_{NF \times NLC}$ = matrix of change in member forces due to change in one member area

$\Delta S_{NF \times NF}$ = matrix of stiffness coefficients incorporating one change in one member area. All other coefficients = zero.

$\Delta \varepsilon_{NF \times NLC}$ = matrix of changes of internal deformation, due to change in one member area.

$\Delta X_{NP \times NLC}$ = matrix of changes in external deformations due to a change in one member area

Then
$$P = A(F + \Delta F) \tag{3.48}$$

$$(F + \Delta F) = (S + \Delta S)(\varepsilon + \Delta\varepsilon) \tag{3.49}$$

$$(\varepsilon + \Delta\varepsilon) = A^T(X + \Delta X) \tag{3.50}$$

and

$$(F + \Delta F) = (S + \Delta S)A^T(X + \Delta X) \tag{3.51}$$

$$P = A(S + \Delta S)A^T(X + \Delta X)$$

Let

$$\Delta K = A\Delta S A^T \tag{3.52}$$

$$P = (K + \Delta K)(X + \Delta X) = KX + K\Delta X + \Delta KX + \Delta K\Delta X$$

For the sake of simplicity $\Delta K\Delta X$ <u>is neglected</u> as a small second-order product.

$P = KX + K\Delta X + \Delta KX$ but $KX = P$

$P = P + K\Delta X + \Delta KX$

$0 = K\Delta X + \Delta KX$ $K\Delta X = -\Delta KX$

$$\underline{\Delta X = -K^{-1}.\Delta K.X} \tag{3.53}$$

Expanding Eq. (3.51)

$$F + \Delta F = SA^TX + SA^T\Delta X + \Delta SA^TX + \Delta SA^T\Delta X \tag{3.54}$$

Neglect $\Delta SA^T \Delta X$ as a second order small term, subtract Eq. (3.5) $F = SA^TX$.

$$\underline{\Delta F = SA^T\Delta X + \Delta S \cdot A^TX} \qquad (3.55)$$

Equations (3.53) and (3.55) are approximations because sone second-order small matrix terms were neglected. The Statics matrix A was not affected. If the equilibrium condition is still satisfied, then $\Delta P = A\Delta F$ should be zero.

$$\Delta P = A\Delta F = ASA^T\Delta X + A\Delta SA^TX \quad \text{but} \quad \Delta K = A\Delta SAT$$

$$\Delta P = K(-K^{-1})\Delta KX + \Delta KX$$

$$\Delta P = A\Delta F = -I\Delta KX + \Delta KX = 0 :$$

Thus the equilibrium condition is satisfied and Eq. (3.53) and (3.55) can be used to develop coefficients in the stress constraint inequalities.

Deformation constraints: see Eq. (3.47)

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta X_{ijk} \cdot u_j \quad \begin{matrix} \leq (x_{ik})\text{all} - x^o{}_{ik} + \sum_{j=1}^{NM} \Delta x_{ijk} ; x^o{}_{ik} \geq 0 \\ > (x_{ik})\text{all} - x^o{}_{ik} + \sum_{j=1}^{NM} \Delta x_{ijk} ; x^o{}_{ik} < 0 \end{matrix}\right]\right.\right.$$

$$\left. i = 1, NP \right] \quad \left. K = 1, NLC \right]$$

Find the value of $\sum_{j=1}^{NM} \Delta x_{ijk}$ = total change in deformation xi due to 100% increase in each veriable j at loading condition k

$$\sum_{j=1}^{NM} \Delta X_{ij} = -(K^{-1})\Delta_{kj} \cdot x = -(K^{-1})A\left(\sum_{j=1}^{NM} \Delta S_j\right)A^TX =$$

$$= -(K^{-1})ASA^TX = -(K^{-1})KX = -I \cdot X = -X$$

This result points to the error caused by neglecting the second order terms : deformation cannot be zero when all member areas are increased by 100%. The result still can be used in the deformation constraint inequalities.

Deformation constraint inequalities in final form are:

$$\left[\left[\left[\sum_{j=1}^{NM} \Delta x_{ijk}\, u_j \;\begin{matrix} \le (x_{ik})\mathrm{all} - 2x^{o}_{ik};\; x^{o}_{ik} \ge 0 \\ \\ > (x_{ik})\mathrm{all} - 2x^{o}_{ik};\; x^{o}_{ik} < 0 \end{matrix}\right] i = 1, NP\right] K = 1, NLC\right] \qquad (3.56)$$

where

Δx_{ijk} = deformation of freedom "i" due to a 100% increase in area of member "j" at loading condition "k"

$u_j = 1 + \dfrac{\Delta A_j}{A^{o}_j}$ solution variable for area A^{o}_j

$(x_{ik})\mathrm{all}$ = allowable deformation of freedom "i" at loading condition "k". This is a signed quantity.

x^{o}_{ik} = deformations of freedom "i" in the original system at loading condition "k".

3.7.5 Side Constraints

(i) Limits will be placed on the solution vector as the constraint relationships are more exact closer to the original design point.

- They are arbitrarily established as ± 20% and will be the first set of side constraints.

$$\begin{aligned} U_i &\geq (LB)i \qquad i = 1,NM \\ U_i &\leq (UB)i \qquad i = 1,NM \end{aligned} \tag{3.57}$$

U_i = solution vector (LB)i = 0.80 lower bound on solution variable.

(UB)i = 1.20 upper bound on solution variable.

(ii) Lower limit on member areas. There is a limit on the smallest size of member which can be used in a structure without penalizing it by more expensive details of very thin members.

$$A^o{}_i . u_i \geq A^{oo}{}_i \qquad i = 1,MM$$

$A^o{}_i$ = original member area

$A^{oo}{}_i$ = minimum area permissible.

(iii) Members with identical areas in order to achieve symmetry and detailing advantages. For such members original areas will be assumed identical.

$$A^o_i = A^o_k$$

$$u_i - u_k = 0$$

3.8 LINEAR PROGRAMMING PROBLEM OF STRUCTURAL DESIGN IN MATRIX FORMULATION

Objective function: $Z = [L_j . A_j . \rho]_{1 \times NM} \{u_j\}_{NM \times 1}$

Constraints:

$$\begin{bmatrix} \Delta F \\ \Delta x \\ I \\ I \\ A^o \\ ID \end{bmatrix}_{NR \times NM} . \{u\}_{NM \times 1} + [I]_{NR \times NR} . \{u'\}_{MR \times 1} = \begin{Bmatrix} F' \\ x' \\ U_B \\ L_B \\ A^{oo} \\ 0 \end{Bmatrix} \begin{array}{l} \text{STRESS CONSTRAINTS} \\ \text{DEFORMATION CONSTRAINTS} \\ \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{SIDE CONSTRAINTS} \end{array}$$

NX = number of designated joints with deformation constraints

$[\Delta F]_{NM \times NM}$, $\{F'\}_{NM \times 1}$ = coefficients in stress constraint inequalities

$[\Delta X]_{NX \times NM}$, $\{X'\}_{NX \times 1}$ = coefficients in deformation constraint inequalities

$[I]_{NM \times NM}$ = identity matrix

$\{UB\}_{NM \times 1}$, $\{LB\}_{NM \times 1}$ = upper and lower limits on solution vector for one cycle

$[A^o]_{NM \times NM}$ = diagonal matrix of original member areas

$\{A^{oo}\}_{NM \times 1}$ = minimum member areas

$[ID]_{NR' \times NM}$ = coefficients of the matrix in identical area constraint equations

NR' = number of equations for identical area constraints

NR = total number of constraint equations

$\{U\}_{NM \times 1}$ = vector of solution variables

$[I']_{NR \times NR}$ = diagonal matrix giving signs to slack variables

$\{U'\}_{NR \times 1}$ = vector of slack variables

3.9 COEFFICIENTS IN MINIMUM TRUSS WEIGHT EQUATIONS

Coefficients ΔX and ΔF – changes in deflections and member forces due to 100% increase of one member's area – are compiled by the program MINTWC from Ref. [1]

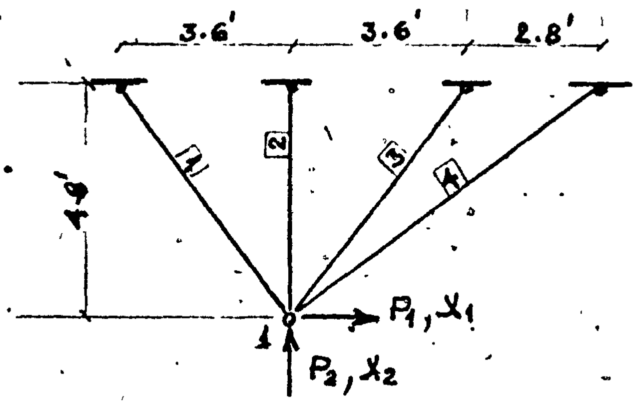
It solves equations:

$$\Delta X = - K^{-1} . \Delta K \quad X \tag{3.53}$$

$$\Delta F = SA^T \Delta X + \Delta S \; A^T X \tag{3.55}$$

These equations are approximate as are all differential operations, because of neglecting the second order small quantities. In this case it was assumed that $\Delta K \Delta X = 0$ and $\Delta S\ A^T \Delta X = 0$.

In order to test the accuracy of the numerical results of MINTWC a direct comparison was made. The 4 bar truss of [1] was used. First, member forces, deformations and coefficients ΔX and ΔF were computed by MINTWC. From these results total deformations X' and total number forces F' were calculated for the truss with area of number 1 increased by 100%. Then the same truss (with area of member 1 increased by 100%) was analyzed directly by DEM and the resulting deformations and member forces compared with X' and F'. Example of 4 bar truss follows:



$A_1 = 1.44\ in^2$. $A_2 = 2.00\ in^2$

$A_3 = 1.80\ in^2$ $A_4 = 2.00\ in^2$

Loading Cond. 1 $P_1 = 0$ $P_2 = 10k$

Loading Cond. 2 $P_1 = 5k$ $P_2 = -6k$

Analyzing by MINTWC.

Deformations in order of loading conditions:

x_{11} = 1.7720496 E-04 x_{12} = 6.138024 E-0

x_{21} = -4.2205268 E-04 x_{22} = -3.4183409 E-04

Change in deformation due to increase of member area 1 by 100%: Al = 2.88 in^2.

ΔX_{111} = -2.399775 E-04 ΔX_{211} = -3.468859 E-04

ΔX_{211} = 1.41915 E-04 ΔX_{212} = 2.051381 E-04

Total deformation by adding deformation and change.

$x'_{11} = x_{11} + \Delta x_{111}$ = -0.6277254 E-04;

$x'_{21} = x_{21} + \Delta x_{211}$ = -2.8013768 E-04;

$x'_{12} = x_{12} + \Delta x_{112}$ = 2.669165 E-04

$x'_{22} = x_{22} + \Delta x_{212}$ =-1.36696-04.

Analyzing by DEM:

Total Deformation. When Al = 2.88 in^2 (increase by 100%)

x'_{11} = 0.25324638 E-04 x'_{12} = 3.9426039 E-04

x'_{21} =-3.32223513 E-04 x'_{22} =-3.1200340 E-04

Differences are considerable. The total deflection as computed by MINTWC is smaller. This was to be expected. The simplification in matrix differentiation assuming

$$\Delta k \times \Delta x = 0$$

was accepting the condition that if all member areas are doubled then all deformations are zero.

Analysing by MINTWC.

Member forces in order of loading conditions:

F_{11} = 3.1965k	F_{12} = 4.6206
F_{21} = 5.2757k	F_{22} = 4.2729
F_{31} = 2.0819k	F_{32} = 0.8533
F_{41} = 0.8360k	F_{42} = 2.1446

Changes in member forces due to increase of member area 1 by 100%:

ΔF_{111} = 1.3424k	ΔF_{112} = 1.9464
ΔF_{211} =-1.7339k	ΔF_{212} =-2.5642
ΔF_{311} = 0.2741k	ΔF_{312} = 0.3962
ΔF_{411} = 0.8012k	ΔF_{412} = 1.1582

The total member forces when A1 = 2.88 in^2 (increase by 100%)

$$F' = F + \Delta F$$

F'_{11} = 4.5389k	F'_{12} = 6.5670k
F'_{21} = 3.3018k	F'_{22} = 1.2087k
F'_{31} = 2.3560k	F'_{32} = -0.4571k
F'_{41} = 1.6372k	F'_{42} = -0.9864k

Equilibrium check at Joint 1 for loading cond. 1 :

$$4.5389 \times 0.80 = 3.63112k$$
$$3.3018 \times 1.00 = 3.5012k$$
$$2.3560 \times 0.80 = 1.8848k$$
$$1.6372 \times 0.60 = 0.9822k$$
$$10.0000 \times 1.00 = -10.0000k$$

$$\Sigma Y = -0.0006k \simeq 0.0k$$ equilibrium is satisfied.

Analyzing by DEM.

Total member forces when A1 = 2.88 in^2 (increase by 100%)

$F'_{11} = 4.0462k$	$F'_{12} = 5.8487k$
$F'_{21} = 4.1529k$	$F'_{22} = 2.6500k$
$F'_{31} = 2.2553k$	$F'_{32} = -0.6026k$
$F'_{41} = 1.3431k$	$F'_{42} = -1.4115k$

Considerable difference in member forces by MINTWC and DEM, although the equilibrium is satisfied.

For both stress and deformation constraints restrictions on change in solution variables are clearly indicated.

CHAPTER 4

DESCRIPTION OF EXISTING COMPUTER PROGRAMS USED AND THEIR ADAPTATION TO THE MINIMUM WEIGHT DESIGN OF A TRUSS

CHAPTER 4

DESCRIPTION OF EXISTING COMPUTER PROGRAMS USED AND THEIR ADAPTATION TO THE MINIMUM WEIGHT DESIGN OF A TRUSS

Three existing computer programs were used for this study with various degrees of modifications:

1) The analysis program (see also Section 3.2) was DEM - Direct element method of truss analysis by Wang [1];

2) MINTWC - Fortran program for coefficients in minimum truss weight equations. By Prof. C.K.Wang [1].

3) LINPROG - Fortran program solving linear programming problem of minimization for a given set of constraint equations and objective function by the Simplex method. Program was written using outline in [3] as an assignment in course CE 653 - "Structural Synthesis", given by Dr. Fang in 1970.

First, these programs were used in their original form to solve the practical problem described in Chapter 5 to test their suitability.

The preliminary design was done for the 2-span truss with five different depths. For testing the MINTWC and LINPROG, a reduced optimization problem was run once for a 15'-0 deep truss. All this was quite possible to do but it also proved without doubt, that the manual calculations involved in redesigning members for repeated analysis cycles and compiling and key punching data cards for optimization in LINPROG were excessive and therefore impractical. Additions and modifications were clearly in order to make the whole design process run automatically. The following is a short description of the work done to these programs.

4.1 DEM-DIRECT ELEMENT METHOD OF TRUSS ANALYSIS

In its original form the program requires two sets of member data. The program was altered so that only one set of member data is required to input.

The correctness of the program was checked by running it with known data and results. Examples in C.K. Wang's book "Matrix Methods in Structural Analysis" were used and the results were identical. This was accepted as sufficient proof that the program DEM in its original format and also with the modifications is correct.

The DEM program was used in the preliminary design of the two-span truss in the practical example. This truss is externally and internally statically indeterminate. For the first analysis, member areas were assumed. For the resulting member forces new sections were designed according to C.S.A. S16 satisfying stress and slenderness requirements. These structures were then re-analyzed and again new member areas were chosen. In about 3 cycles there was no more changes in the member sizes and this was assumed to be the optimum arrangement for the depth of the truss chosen.

This was repeated for the trusses with depth 20'-0,15'0, 7'-6",5'-0 and 3'-0, relationship between the weight of the truss and the depth of the truss displayed in a graph and optimum depth - with smallest weight - 6'-0 ± was selected by inspection.

Using this program required a considerable amount of manual work, therefore DEM was further altered.

DEM was changed to DEMTDP to be used in preliminary design to produce five designs, varying the depth automatically. It was arranged exactly the same way as in the manual design. From input of preliminary member areas and loading conditions member forces are computed and new discrete members are chosen from a list according to CSA S16 satisfying all strength and slenderness requirements. Then the structure is again analyzed with the new member areas and the cycle is repeated until the weight of the truss does not change by more than 1% between consecutive cycles. If allowable deformations are exceeded, all member areas are upgraded in the same ratio.

This design sequence is repeated for various depths of truss. At present, the program will produce designs for five different truss depths (2 span truss - see Chapter 5) in about 15 seconds of CPU time.

Thus, with DEMTDP, the preliminary design is fully automatic except for choosing the optimal depth from the various designs. The graph still has to be drawn manually with the added influences of the cost of the wall, cost of the columns, cost of the approachs, etc., included. This also can be done by computer but the additional programming effort was considered out of range for this study.

4.2 MINTWC

This program in its original form generates changes in the member forces caused by doubling the areas of the members one at a time. From these changes the coefficients for the stress constraint equation can be compiled. Originally, it was applicable to a structure of four members only and apparently was designed directly for the sample problem in [1] - a 4-bar truss.

a) MINTWC was altered to be applicable to a truss of any number of members and also made to generate the coefficients required for compiling the deformation constraint equations. Originally, this program required submission of member data four times and was modified to reading member data once only.

MINTWC in this form was used once to generate coefficients for a shortened version of optimization of the 15'-0 deep truss of preliminary design. The coefficients for constraint equations were compiled manually from the data generated by MINTWC and data cards were prepared for using in LINPROG. Only one optimization cycle was completed. It worked well and the convergence with LINPROG was fast but time spent on preparing the data even for one cycle was excessive. Considering that complete optimization might take several cycles and data would have to be prepared each time until there is less than 0.5% weight difference between two consecutive designs it was obviously impractical.

b). MINTWC was changed into MINTW to compile the coefficients directly in a form acceptable to LINPROG.

4.3 LINPROG

This program was not changed except for minor modification when it was used as a subroutine and the data was not submitted with the data cards but read directly from memory as it was compiled by the Program MINTW.

4.4 USE OF THE COMPUTER PROGRAMS IN OPTIMIZATION

These programs were used in solving the optimization problem in two different ways :

a) by assuming that member areas are available in gradually varying sizes and the allowable compressive stress remains constant during the design process. These are the assumptions made in [1] and [5]. It can be considered reasonable if the preliminary design data is close to optimal and only small changes are anticipated. Even so much of the computed optimality is lost at the end while choosing the nearest available section for the optimal member's area. Its main advantage is computational efficiency.

b) by using discrete member sizes and choosing a new buckling stress - allowable compressive stress - after each optimization cycle. This increases the computational steps but appears to match better with the actual structural design process.

Both procedures were tried by combining the available programs.

4.4.1 Gradually Varying Areas and Constant Allowable Compressive Stress

In this form MINTW was used as the main program combined with LINPROG and DEM as subroutines.

MINTW receives as input on data cards member data and allowable compressive stresses and deflections from a preliminary design. MINTW compiles coefficients for constraint equations and objective functions in primal form. The linear programming problem is solved by LINPROG. The optimized areas are adjusted for any constraint violations by DEM. Then the weight of the truss is computed and compared with the weight of the truss from the previous cycle. If the difference in weight is more than 0.5% the cycle is repeated until the difference in weight becomes smaller than 0.5% or the prescribed maximum number of cycles is completed. The result is then accepted as optimal.

4.4.2 Member Sizes From a List

With the member sizes selected from a list the preliminary design is combined with the final design.

As the main program is used DEMTD with subroutines MINTW, LINPROG and DEMTR, where DEMTR is a slightly modified version of DEMTD. DEMTD is identical to DEMTDP except that it designs a truss of one depth. (Optimal depth from preliminary design).

DEMTD receives as input the truss geometry only.

The convergence is better if the member areas are close to the optimum but unit values would work also. Allowable compressive stress is calculated while discrete member is selected following the same code requirements as in the preliminary design. The resulting areas and compressive stresses are then used by MINTW to produce the coefficients for the constraint equations and objective function. The linear programming problem is solved by LINPROG and optimal areas adjusted by DEMTR again selecting discrete sections from the list. These new discrete member areas and new allowable compressive stresses are used by MINTW and the cycle is repeated until the difference in weight of the truss between the successive cycles is less than 0.5% or the prescribed maximum number of cycles is completed. The discrete sections from the last cycle are then the final optimal ones.

This last procedure with discrete member sizes and preliminary design combined with final design is considered to be the better of the two alternatives for optimal design of a truss.

4.5 LIMITATIONS OF THE PROPOSED METHOD

DEMTD with subroutines has some limitations, which can be removed by more programming work.

1) All loading combinations have to be directly input as there are no provisions for adding up any pre-

scribed loading cases. The program only shows max. or min. member forces from the loading cases given as input and designs a member for it.

2) At the present time, the list of discrete members is limited and includes only double angles with 3/8" spacing.

3) The change in dead weight of the truss is not included in the loads after each cycle as it was considered small compared with the total loads.

4) Optimization in LINPROG takes much computer time. For the general case of many loading conditions and correspondingly large numbers of constraint equations, it would be less time-consuming to solve the dual problem instead of the primal.

CHAPTER 5

PRACTICAL EXAMPLES - 2 SPAN TRUSS

CHAPTER 5

PRACTICAL EXAMPLE - 2 SPAN TRUSS

This problem was solved in two different ways in order to demonstrate the use of the existing programs in their various stages of development.

1) DEM, MINTWC, LINPROG were used in their original forms (with only small changes) as independent programs. All the necessary calculations and data preparation for linking them together were done manually. New data cards were prepared for each program. As expected, this arrangement uses much of the designer's time and comparatively little computer time, and is not really an automatic design sequence. However, it checked out the component programs and clearly established the need for automatization. In this scheme, it is assumed that the allowable compression stress remains constant during the optimization process in MINTWC and LINPROG and the member areas are available in gradually varying sizes.

2) DEMTD, MINTW, LINPROG, DEMTR with modifications were linked together as a program with subroutines and only one data submission.

Preliminary design to establish the optimal depth is done by DEMTDP.

The optimal geometry with member areas from the nearest preliminary design are submitted to DEMTD with subroutines MINTWC, LINPROG and DEMTR. In this scheme new discrete members are chosen from a list after each cycle of optimization and new compressive stress (buckling stress) is chosen for each new cycle.

This process is fully automatic except for choosing the optimal depth from the various preliminary designs which still have to be done by inspection from a graph showing the relationship between the truss depth and the weight.

Problem statement, assumptions for design, mathematical model and loading conditions are the same for both procedures 1) and 2), and are stated only once.

Problem Statement

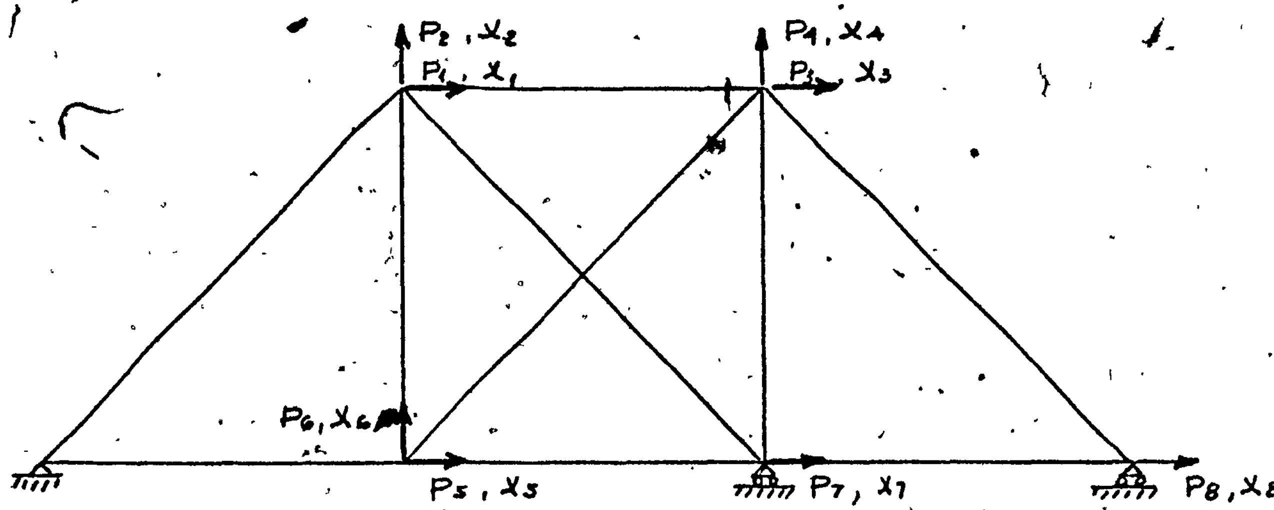
Design a two-span truss for given loads with spans 30' and 15' choosing the depth of the truss and member areas so that the total weight of the truss is minimum. The depth of the truss does not influence any other cost items.

Assumptions for Design

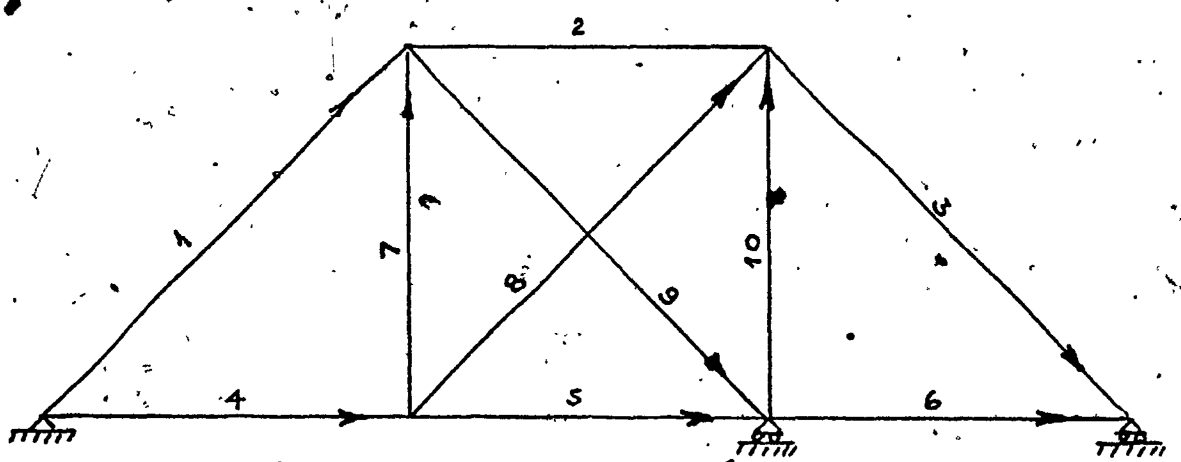
Material - steel F_y = 44 ksi (CSA G40.12)

Design specifications - NBC 1970 and CSA S16-1969

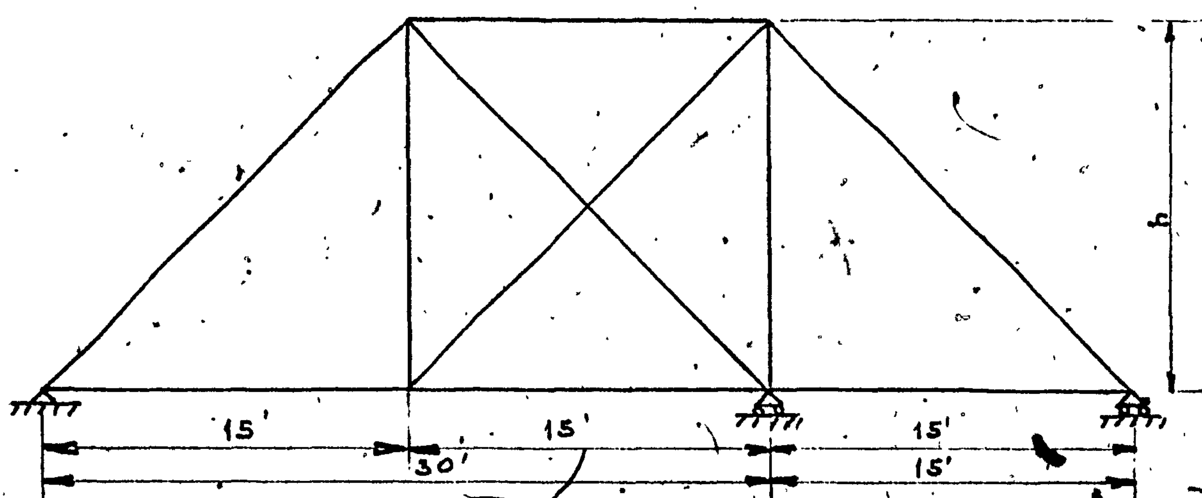
Connections - welded



P AND X DIAGRAM



MEMBER NUMBERS, INITIAL AND TERMINAL ENDS



DIMENSIONS

2 SPAN TRUSS - MATHEMATICAL MODEL

Minimum area of a member 2.12 sq. in (2 ⫪ $2\frac{1}{2} \times 2 \times \frac{1}{4}$")

Weight of steel 490 lbs/ft^3 = 0.284 lbs/in^3

Deformations restricted on freedoms P_3 and P_6 with

x allow = 0.18"

Identical member areas : A1 = A3, A4 = A5, A8 = A9.

External Loads

a) Dead load: Deck and bracing $P_2 = P_4 = P_6 = -1K$

Weight of a truss with depth h = 15'-0" is computed from assumed areas:

$A_1 = A_3 = 4$ in^2

$A_2 = 5$ in^2

$A_4 = A5 = 5$ in^2

$A_6 = 5$ in^2

$A_7 = 2$ in^2

$A_8 = A9 = 4$ in^2

$A_{10} = 2$ in^2

$$L_1 = L_3 = L_8 = L_9 = 15/0.707 = 21.2' = 254''$$

$$L_2 = L_4 = L_5 = L_6 = L_7 = L_{10} = 15' = 180''$$

Joint loads due to weight of truss (10% added for connections).

$$P_2 = -\tfrac{1}{2}(245 \times 4 \times 180 \times 5 \times 180 \times 2 \times 254 \times 4) \times \frac{490}{12^3} \times$$

$$\times 1.10 = -511 \text{ lbs}$$

$$P_4 = P_2 = -511 \text{ lbs}$$

$$P_6 = -\tfrac{1}{2}(180 \times 5 \times 180 \times 2 \times 254 \times 4 \times 180 \times 5) \times \frac{490}{12^3} \times$$

$$\times 1.10 = -450 \text{ lbs}$$

Total weight of truss:

$$W = (2 \times 254 \times 5 + 4 \times 180 \times 5 + 2 \times 180 \times 2 +$$

$$2 \times 254 \times 4) \times 0.284 \times 1.10 =$$

$$= 2299 \text{ lbs}$$

b) Live Load: LC_1' $P_6 = -40$ k

LC_2' $P_2 = -40$ k

LC_3' $P_4 = -40$ k

LC_4' $P_1 = +20$ k $P_3 = +20$ k $P_4 = -30$ k

$P_5 = +5.0$k $P_6 = -30$ k $P_7 = +50$ k

$P_8 = +5.0$k

<u>Loading conditions for design</u> are live load with dead load of deck, bracing and truss added.

LC_1 $P_2 = -(1 + 0.511) = -1.511$ k

$P_4 = -(1 + 0.511) = -1.511$ k

$P_6 = -(1 + 0.495 + 40.0) = -41.495$ k

LC_2 $P_2 = -(1.511 + 40) = -41.511$ k

$P_4 = -1.511$ k

$P_6 = -1.495$ k

LC_3 $P_2 = -1.511$ k

$P_4 = -(1.511 + 40) = -41.511$ k

$P_6 = -1.495$ k

LC_4 $P_1 = -20.0$ k

$P_2 = -1.511$ k

$P_3 = +20.0$ k

$P_4 = -(1.511 + 30) = -31.511$ k

$P_5 = +5.0$ k

$P_6 = -(1.495 + 30.0) = -31.495$ k

$P_7 = +5.0$ k

$P_8 = +5.0$ k

Although the weight of the truss will be different for different truss depths, the weight of a 15'-0 deep truss is used as dead load for all other cases. This variation is expected to be small in this case, compared to the total DL + live joint loads.

5.1 A: DEM, MINTWC, LINPRQG USED IN THEIR ORIGINAL FORM

With this procedure, trusses of the depths 20', 15', 7'-6", 5' and 3'-0 were designed.

For given loads, geometry and assumed member areas the truss is analyzed by DEM and for the resulting member forces new members are designed. As this is a statically indeterminate truss, new member areas will influence the member forces. Therefore, the truss is reanalyzed with the new areas and for the new distribution of forces the members are designed again. This cycle is repeated until the difference in the truss weights between the last two consecutive designs is less than 1%. This is, then, assumed to be the optimal weight for that depth.

In this example there is no change in the member areas and truss weights between the second and third analysis. So the areas of the 3rd analysis are assumed as optimal for preliminary design of that depth.

Following are the results of the analysis and design for the truss of depth h = 15' arranged in Tables 1 to 6.

TABLE 1

FIRST ANALYSIS h = 15'-0

Member No.	Member Forces in Order of Loading Conditions				Critical Member Forces		h=15'
	LC_1	LC_2	LC_3	LC_4	Max. kips	Min. kips	
1	-29.4167	-29.3034	- 9.0416	- 3.2046	--	-29.4167	
2	-12.6073	5.3376	- 3.3170	- 5.6115	--	-12.6073	
3	1.9863	2.2130	-13.8320	-16.3003	2.213	-16.3003	
4	20.8007	20.7206	6.3933	57.26620	56.2660	--	
5	6.7889	13.8182	12.8571	38.1806	38.1806	--	
6	- 1.4040	- 1.5648	9.7807	16.5260	16.3260	- 1.5648	
7	27.4832	- 5.4074	7.9587	17.4095	27.4832	- 5.4074	
8	-11.5873	-21.7548	- 4.3507	-23.5531	--	-23.5531	
9	19.8158	9.7615	- 9.1411	19.9198	19.8158	- 9.1411	
10	-16.9274	- 9.9782	-23.2666	-34.0704	--	-34.0704	

TABLE 2

FIRST DESIGN $h = 15'-0$

	Member Forces		Area	L	$R_{min} = \frac{L}{300}$ in	$R_{min} = \frac{L}{200}$ in	New Section						Weight of Member lb.	Fa ksi
	Max. kips	Min kips	in^2	Feet			Mark	Weight lb/LF	Area in^2	r in	P Allowable Tension + k	P Allowable Compr. - k		
1	--	-29.4167	4.0	21.2	---	1.27	2 ⌉⌈ 3 x 3½ x 7/16	24.0	8.0	1.49	---	-42.2	509	---
2	--	-12.6073	5.0	15.0	---	0.90	2 ⌉⌈ 3½ x 2½ x 5/16	11.2	3.56	1.10	---	-17.4	168	---
3	2.2130	-16.3003	4.0	21.2	---	1.27	2 ⌉⌈ 3½ x 2½ x 5/16	15.2	3.56	1.10	---	-17.4	237	---
4	57.2660	--	5.0	15.0	0.60	--	2 ⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	72.6	---	150	---
5	38.1806	--	5.0	15.0	0.60	--	2 ⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	72.6	---	150	---
6	16.5260	- 1.5648	5.0	15.0	---	0.90	2 ⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	72.6	-10.94	150	-3.73
7	27.4832	- 3.4074	2.0	15.0	---	0.90	2 ⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	72.6	-10.94	150	---
8	---	-23.5531	4.0	21.2	---	1.27	2 ⌉⌈ 3 x 3½ x 7/16	24.0	7.06	1.49	72.6	-38.2	509	---
9	19.9198	- 9.1411	4.0	21.2	---	1.27	2 ⌉⌈ 3 x 3½ x 7/16	24.0	7.06	1.49	---	----	509	-4.98
10	---	-34.0704	2.0	15.0	---	0.90	2 ⌉⌈ 4 x 3	19.6	5.74	1.25	---	-37.2	294	-6.31

$$\begin{array}{lrl} & & 2686 \text{ lbs} \\ \text{connections } 10\% & = & \underline{269} \\ & & 2955 \text{ lbs} \end{array}$$

$$P_2 = -(W_1 + W_2 + W_7 + W_8) \times 0.55 = -652 \text{ lbs}$$

$$P_4 = -(W_2 + W_3 + W_9 + W_{10}) \times 0.55 = -586 \text{ lbs}$$

$$P_6 = -(W_4 + W_5 + W_7 + W_9) \times 0.55 = -430 \text{ lbs}$$

TABLE 3

SECOND ANALYSIS h = 15'-0

Member No.	Member Forces in Order of Loading Conditions				Critical Member Forces		h=15'
	LC_1	LC_2	LC_3	LC_4	Max. kips	Min. kips	
1	-27.2591	-27.6466	- 4.7983	5.7003	5.7003	-27.6466	
2	-11.0551	- 4.5948	- 0.5258	- 0.2369	--	-11.0551	
3	6.4372	5.6623	- 5.2098	1.6454	6.4372	- 5.2098	
4	19.2751	19.5491	3.3929	50.9693	50.9693	--	
5	3.6882	10.9504	6.5510	24.5689	24.5689	--	
6	- 4.5518	- 4.0038	2.6839	3.8365	3.8365	- 4.5518	
7	25.8431	- 7.1486	4.6081	10.0496	25.8431	- 7.1486	
8	-11.6249	-21.1485	- 4.0547	-22.2489	--	-22.2489	
9	22.0715	12.1603	- 4.4662	30.2647	30.2647	- 4.4662	
10	-21.7447	-14.7491	-34.7441	-54.1498	--	-54.1498	

TABLE 4

SECOND DESIGN h = 15'-0

	Member Forces		Area	L	$R_{min} = \frac{L}{300}$ in	$R_{min} = \frac{L}{200}$ in	New Section						Weight of Member	Fa
											P Allowable			
	Max kips	Min kips	in²	ft			Mark	Weight lb/ft	Area in²	r in	Tension + k	Compr. - k	lb.	ksi
1	5.7003	-27.646	8.0	21.2	---	1.27	2⌉⌈3 x 3½ x 7/16	24.0	7.06	1.47	187.0	- 35.1	509	-4.58
2	---	-11.0551	3.56	15.0	---	0.90	2⌉⌈3 x 3½ x 2½ x 5/16	11.2	3.56	1.10	94.0	- 19.70	168	-5.54
3	6.4372	- 5.2098	3.56	21.2	---	1.27	2⌉⌈3 x 3½ x 5/16	24.0	7.06	1.47	187.0	- 35.1	509	-4.98
4	50.9693	---	2.94	15.0	0.60	---	2⌉⌈3 2½ x 2 x ¼	7.24	2.12	0.78	56.0	---	109	---
5	24.5689	---	2.94	15.0	0.60	---	2⌉⌈ 2½ x 2 x ¼	7.24	2.12	0.78	56.0	---	109	---
6	3.6839	4.5318	2.94	15.0	---	0.90	2⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	77.6	- 9.84	150	-3.35
7	25.8431	- 7.1486	2.94	15.0	---	0.90	2⌉⌈ 3 x 2 x 5/16	10.0	2.94	0.90	77.6	- 9.84	150	-3.35
8	---	-22.2488	7.06	21.2	---	1.27	2⌉⌈3 x 3½ x 7/16	24.0	7.06	1.47	187.0	- 35.1	509	-4.98
9	30.2647	- 4.466	7.06	21.2	---	1.27	2⌉⌈3 x 3½ x 7/16	24.0	7.06	1.47	187.0	- 35.1	509	-4.98
10	---	-54.1498	5.74	15.0	---	0.50	2⌉⌈3 x 3½ x 7/16	24.0	7.06	1.47	---	- 69.5	360	-9.85

connections 10% = $\frac{3082 \text{ lbs} + 308}{3390 \text{ lbs}}$

$P_2 = -(W_1 + W_2 + W_7 + W_8) \times 0.55 = -735$ lbs

$P_4 = -(W_2 + W_3 + W_9 + W_{10}) \times 0.55 = -930$ lbs

$P_6 = -(W_4 + W_5 + W_7 + W_9) \times 0.55 = -430$ lbs

TABLE 5

THIRD ANALYSIS h = 15'-0

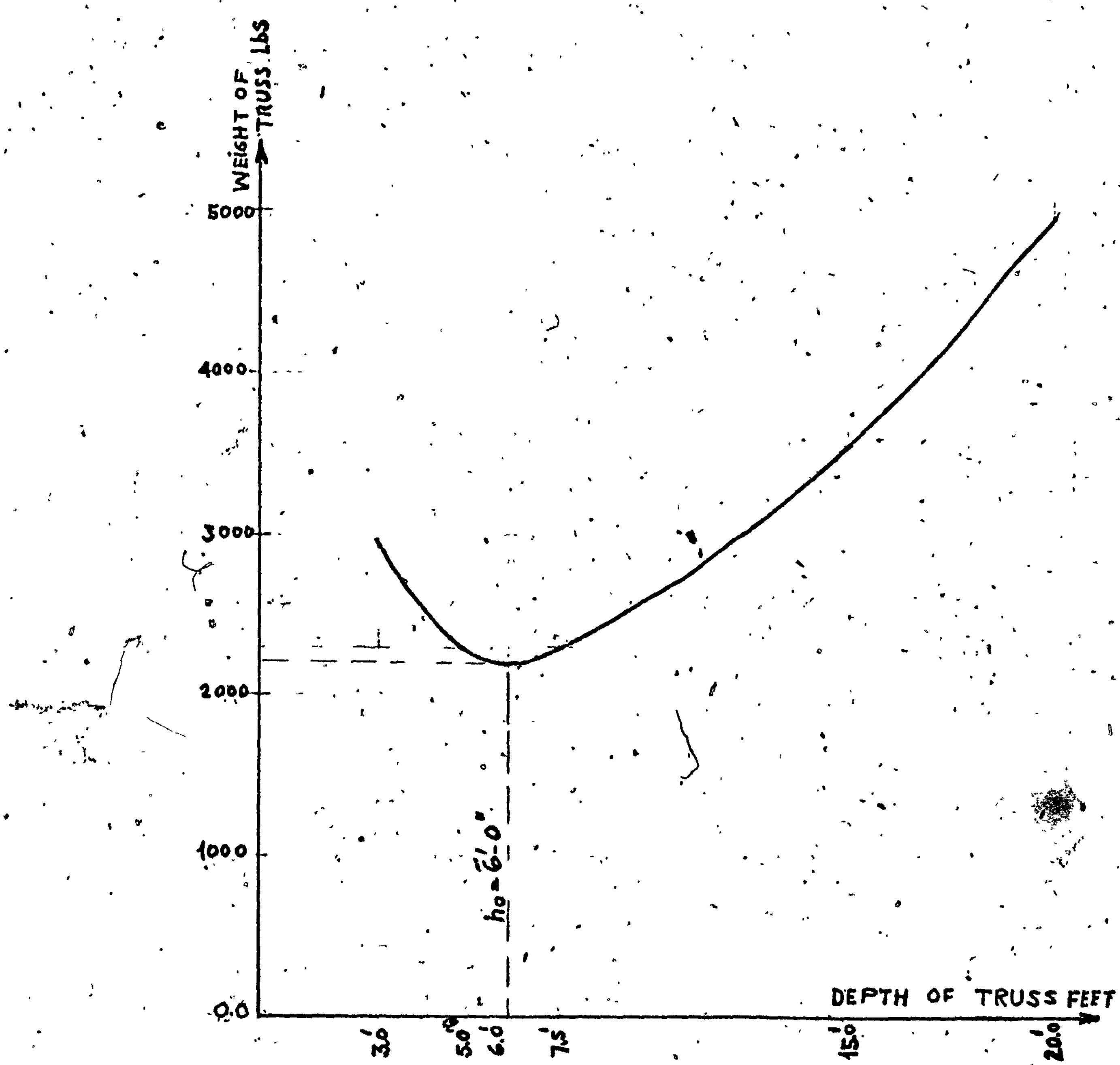
Member No.	Member Forces in Order of Loading Conditions				Critical Member Forces		h=15'
	LC_1 (k)	LC_2 (k)	LC_3 (k)	LC_4 (k)	Max. kips	Min. kips	
1	-25.4212	-25.7286	- 4.6642	8.8679	8.8679	-25.7286	
2	- 9.1604	- 2.8895	- 1.2258	1.6619	1.6619	- 9.1604	
3	10.2772	9.6622	- 4.7775	8.1446	10.2772	- 8.1446	
4	17.9753	18.1929	3.2981	48.7294	48.7294	---	
5	1.5480	8.4712	5.4505	19.6322	19.6322	---	
6	- 7.2671	- 6.8322	3.3782	- 0.7591	3.3782	- 8.2671	
7	25.0555	- 8.2387	3.6354	7.3858	25.0555	- 8.2387	
8	-12.4663	-21.6423	- 2.9307	-21.7666	---	-21.7666	
9	23.2320	13.7485	- 3.0480	34.0787	34.0787	- 3.0440	
10	-25.6246	-18.4839	-36.3994	-61.7863	---	-61.7863	

TABLE 6

THIRD DESIGN h = 15'-0

	Member Forces Max kips	Member Forces Min kips	Area in^2	L ft	$R_{min} = \frac{L}{300}$ in	$R_{min} = \frac{L}{200}$ in	New Section Mark	Weight lb/ft	Area in^2	r in	P Allowable Tension + k	P Allowable Compr. -k	Weight of Member lb.	Fa ksi
1	8.8679	-25.7286	7.06	21.2	---	1.27	2┐┌5 x 3½ x 7/16	24.0	7.06	1.47	187.0	-35.1	509	-4.98
2	1.6619	- 9.1604	3.56	15.0	---	0.90	2┐┌3½ x 2½ x 5/16	11.2	3.56	1.10	94.0	-19.70	168	-5.54
3	10.2772	- 4.7795	7.06	21.2	---	1.27	2┐┌5 x 3½ x 5/16	24.0	7.06	1.47	187.0	-35.1	509	-4.98
4	48.7294	---	2.12	13.0	0.60	---	2┐┌2½ x 2 x ¼	7.24	2.12	0.78	56.0	---	109	---
5	19.6322	---	2.12	15.0	0.60	---	2┐┌2½ x 2 x ¼	7.24	2.12	0.78	56.0	---	109	---
6	3.3782	- 7.2671	2.94	15.0	---	0.90	2┐┌3 x 2 x 5/16	10.0	2.94	0.50	77.6	- 9.84	150	-3.35
7	25.0555	- 8.2387	2.94	15.0	---	1.27	2┐┌3 x 2 x 5/16	10.0	2.94	0.90	77.6	- 9.84	150	-3.35
8	---	-21.7666	7.06	21.2	---	1.27	2┐┌5 x 3½ x 7/16	24.0	7.06	1.47	187.0	-35.1	509	-4.98
9	34.0787	- 3.044	7.06	21.2	---	1.27	2┐┌5 x 3½ x 7/16	24.0	7.06	1.47	187.0	-35.1	509	-4.98
10	---	-61.7863	7.06	15.0	---	0.90	2┐┌5 x 3½ x 7/16	24.0	7.06	1.49	---	-69.5	360	-9.85

connections 10% = 3082 lbs
308
3390 lbs



The Relationship Between the Depth of Truss and the Weight of Truss - Truss Analysis by DEM- Minimum Weight of Truss is at Depth Approximately 6'-0"

GRAPH NO. 1

Identical sets of analyses and designs were done for trusses with depths 20'-0, 7'-6", 5'-0" and 3'-0" with the following results:

h = 20'	Optimal weight of truss	W = 4940 lbs
h = 15'		W = 3390 lbs
h = 7'-6"		W = 2310 lbs
h = 5'-0		W = 2301 lbs
h = 3'-0		W = 2971 lbs

These results are displayed in Graph No. 1. showing the relationship between the weight and the depth of trusses. Optimal depth was chosen by inspection to be

$$h = 6'-0$$

It was assumed that the depth of the truss did not affect any other cost factors in this particular case. See also problem statement.

Finding optimal depth this way involves a considerable amount of manual calculation. For each depth, approximately three analyses and three design cycles are required. This means redesigning the truss 15 times. Analysis by DEM is fast but the data still has to be prepared and submitted 15 times. That is for most cases unreasonably expensive in designer's time although savings in weight can be large;

Redesigning the Truss for Optimum Depth $h = 6'-0$

1st Cycle, Areas in^2:

$A_1 = 7.06$ $A_2 = 2.34$ $A_3 = 7.06$ $A_4 = 2.62$ $A_5 = 2.62$

$A_6 = 4.96$ $A_7 = 2.12$ $A_8 = 7.06$ $A_9 = 7.06$ $A_{10} = 4.96$

2nd Cycle, Areas in^2:

$A_1 = 7.06$ $A_2 = 2.94$ $A_3 = 7.06$ $A_4 = 2.62$ $A_5 = 2.62$

$A_6 = 5.30$ $A_7 = 2.12$ $A_8 = 6.50$ $A_9 = 6.50$ $A_{10} = 3.46$

Optimal weight $W = 2300.0$ lbs.

Comparing weights of trusses with depth 20' and optimal:

$h = 20'$ $W = 4940$ lbs

$h = 6'$ $W = 2300$

$$W_{20} - W_6 = 2640 \text{ lbs}$$

$$\text{Savings } \Delta W = \frac{2640}{4940} \times 100 = 53.5\%$$

To demonstrate the use of MINTWÇ, LINPROG and DEM for optimization in final design the truss of depth $h = 15'-0$ was chosen. Member areas and allowable compressive stresses are taken from the preliminary design.

The coefficients ΔF and Δx for compiling constraint inequalities for the linear programming problem are generated by MINTWC. But coefficients for the objective function and also for constraints have to be calculated manually. See Table 7 P. 90.

This procedure on structural design is based on linear programming. See also Section 3.7. The form of the objective function, stress and deformation constraints for the 2-span truss with 10 members and eight freedoms is as shown.

Objective function:

$$\text{Min } Z = (A_1^o L_1 U_1 + A_2^o L_2 U_2 + A_3^o L_3 U_3 + \ldots\ldots +$$

$$+ A_{10}^o L_{10} U_{10})\ 0.284$$

where L = inches, A = in^2, Z = lbs.

$$L_1 = L_3 = L_8 = L_9 = 254";$$

$$L_2 = L_4 = L_5 = L_6 = L_7 = L_{10} = 180"$$

$$A_1^o = A_3^o = A_8^o = A_9^o = A_{10}^o = 7.06 \text{ in}^2$$

$$= 7.06 \text{ in}^2$$

$$A_2^o = 3.56 \text{ in}^2;\ A_4^o = A_5^o = 2.12 \text{ in}^2;\ A_6^o = A_7^o = 2.94 \text{ in}^2$$

Stress constraints:

$$\Delta F_{i1k}U_1 + \Delta F_{i2k}U_2 + \ldots + [\Delta F_{iik} - (f_{ik})_{all}.A^o_i]\,U_i + \ldots$$

$$\ldots + \Delta F_{i9k}U_9 + \Delta F_{i10k}U_{10} \leq - F^o_{ik}: \text{ For } F^o_{ik} \geq 0.$$

$$\Delta F_{i1k}U_1 + \Delta F_{i2k}U_2 + \ldots + [\Delta F_{iik} - (f_{ik})_{all}.A^o_i]U_i + \ldots$$

$$\ldots + \Delta F_{i9}\,U_9 + \Delta F_{i10}\,U_{10} > - F^o_{ik}; \text{ For } F^o_{ik} < 0$$

$i = 1,2, \ldots 10 \quad k = 1,2,3,4$

In these equations (f_{ik})all = 26.4 ksi for tension members

(f_{ik})all = buckling stress - from preliminary design for compression members.

Deformation constraints:

$$\Delta X_{i1k}U_1 + \Delta X_{i2k}U_2 + \ldots + \Delta X_{i9k}U_9 + \Delta X_{i10k}U_{10} \geq (X_{ik})_{all} -$$

$$- 2X^o_{ik}$$

For $X^o_{ik} \leq 0$

$$\Delta X_{i1k}U_1 + \Delta X_{i2k}U_2 + \ldots + \Delta X_{i9k}U_9 + \Delta X_{i10k}U_{10} < (X_{ik})_{all} -$$

$$- 2X^o_{ik}$$

For $X_{ik} > 0$

$i = 1,2 \ldots 8 \quad k = 1,2,3,4$

ΔF and ΔX are compiled by MINTWC

Side Constraints:

Upper and lower bounds on solution variables are set as 1.20 and 0.80.

The minimum areas for all members $A_{min} = 2.12\ in^2$

Identical Member Areas

$$A_1 = A_3$$

$$A_4 = A_5$$

$$A_8 = A_9$$

All inequalities are made into equations by using slack variables. The linear programming problem will be solved by the Simplex algorithm with infeasibility function (artificial variables) to start. All equations are brought to a form where the right-hand side is positive.

Units Used:

Input:	E	=	30,000 ksi
	H,V	=	inches
	A	=	in^2
	P	=	kips
Output:	X	=	inches
	F	=	kips
	X	=	inches

ΔX = inches

ΔF = kips

Coefficients $\Delta Fijk$, $\Delta Xijk$, F^oik, Xik are taken from MINTWC directly. $\Delta Fijk$ - (fik)all. A^o_i and (Xik)all - $2X^oik$ will be computed before writing constraint equations. See Table 7, p. 90.

For a 10 member truss and 4 loading conditions, if all stress constraints, deformation constraints and side constraints are applied the linear programming problem becomes quite large and not very well suitable for demonstrating manual data preparation.

The size of the various matrices in the Simplex tableau will be as follows:

Stress constraints	[ΔF] 40 x 10
Deformation constraints	[ΔX] 32 x 10
Upper bounds on solution variables	[I] 10 x 10
Lower bounds on solution variables	[I] 10 x 10
Minimum areas of members	[A_o] 10 x 10
Identical areas of members	[I] 3 x 10

$$NR = 40 + 32 + 10 + 10 + 10 + 3 = 105$$

n = 10 - number of design variables

m = 105 - number of slack variables assuming all constraints to be inequalities

TABLE 7

COMPILING COEFFICIENTS IN STRESS AND DEFORMATION CONSTRAINT INEQUALITIES

	LC	Stress					Deformations			
	K	ΔFiik	(fik)all ksi	A^o_1 in^2	(fik)all x A_i^o	ΔFiik − (fik)all x A_i^o	(Xik)all in	X^o_{ik}	$2X^o_{ik}$	bi = (Xik)all $-2X^o_{ik}$
1	4	.5571	26.4	7.06	186.0	-185.44	.180	.1228	.2256	---
2	4	.4931	26.4	3.56	94.0	- 93.51	.180	.1077	-.2154	---
3	4	2.0468	26.4	7.06	186.0	-183.95	.180	.1256	.2512	.0712
4	4	3.6040	26.4	2.12	560.0	-52.3954	.180	.0525	-.1050	---
5	4	6.3123	26.4	2.12	560.0	- 49.68	.180	.1379	.2758	---
6	4	- .1620	-3.35	2.94	- 9.85	9.69	-.180	-.1228	-.2456	.0656
7	4	1.2382	26.4	2.94	77.5	- 76.26	.180	.1934	.3870	---
8	4	-5.3868	- 4.98	7.06	- 3.51	29.21	.180	.1993	.3986	---
9	4	6.7292	26.4	7.06	186.0	-179.27	---	---	---	---
10	4	-10.3626	- 9.85	7.06	- 7.02	59.84	---	---	---	---

m' = 105 - number of artificial variables

N columns = 10 + 105 + 105 = 220

Simplex tableau will have 105 + 2 = 107 rows, 220 + 1 = 221 columns, with a basis matrix 105 x 105. That is a large matrix to handle!

To simplify this demonstration problem for manual data preparation, it was reduced by using one loading condition, LC4 only.

Deformations are constrained in freedoms X_3 and X_6 only, but all side constraints are observed.

To reduce the problem further, the condition is considered that some members are specified to have identical areas. By assuming the original areas of those members identical, the optimal design variables will be identical also. That reduces the number of independent design variables by 3.

In this problem:

$A_1 = A_3$, $A_1^o = A_3^o$, therefore $U_1 = U_3$

$A_4 = A_5$, $A_4^o = A_5^o$, therefore $U_4 = U_5$

$A_8 = A_9$, $A_8^o = A_9^o$, therefore $U_8 = U_9$

That leaves only 7 independent variables in this problem and also eliminates 3 constraint equations, specifying the identical member areas. The new design variables will be:

$$U_1' = U_1 = U_3$$

$$U_2' = U_2$$

$$U_3' = U_4 = U_5$$

$$U_4' = U_6$$

$$U_5' = U_7$$

$$U_6' = U_8 = U_9$$

$$U_7' = U_{10}$$

Number of rows in simplex tableau basis:

$$NR = 10 + 2 + 7 + 7 + 7 = 33$$

Number of columns:

$$NCOL = 7 + 33 + 33 = 73$$

Basis matrix for this reduced problem will be of a size 33 x 33. Constraint equations with slack variables added and right-hand side made positive, with objective functions for the linear programming problem as handled by LINPROG are as follows:

Stress constraints:

1) $184.4166U_1'+0.1965U_2'+7.3607U_3'-0.1145U_4'+0.0568U_5'-1.4464U_6'-$
$-4.0796U_7'-1.0U_8' = 8.8679$

2) $-2.1214U_1'+93.51U_2'+5.33U_3'-0.1537U_4'-1.1579U_5'+0.2278U_6'-1.1762U_7'-$
$-1.0U_9 = 1.6619$

3) $182.835U_1'-0.3930U_2'+15.215U_3'-.2290U_4'+0.1136U_5'-2.8927U_6'-$
$-8.1592U_7'-1.0U_{10} = 8.1446$

4) $1.1176U_1'+0.1390U_2'+50.795U_3'+0.0810U_4'-0.0401U_5'+1.0229U_6' +$
$+ 2.8847U_7'-1.0U_{11} = 48.7294$

5) $1.2315U_1'-0.0762U_2'+45.6081U_3'+0.0892U_4'-1.2783U_5'+2.8450U_6' +$
$+ 7.4779U_7'-1.0U_{12} = 19.6321$

6) $-2.2352U_1'-0.2779U_2'+10.4096U_3'+9.6900U_4'+0.0803U_5'-2.0455U_6' +$
$+ 5.7694U_7'-1.0U_{13} = 0.7591$

7) $0.1138U_1'-0.2152U_2'-5.0796U_3'+0.0083U_4'+76.2600U_5'+1.8176U_6' +$
$+ 4.5932U_7'-1.0U_{14} = 7.3858$

8) $-1.4195U_1'-0.5008U_2'+0.1771U_3'-0.1029U_4'-1.6943U_5'+36.2210U_6' +$
$+ 2.4161U_7'-1.0U_{15} = 21.7666$

9) $-0.1611U_1'+0.3043U_2'+7.1837U_3'-0.0117U_4'+1.7511U_5'+183.4287U_6'-$
$- 6.4957U_7'-1.0U_{16} = 34.0787$

10) $-2.3491U_1'-0.0627U_2'+15.5025U_3'-0.1702U_4'+1.3185U_5'-3.8631U_6' +$
$+ 59.8400U_7'-1.0U_{17} = 61.7868$

Deformation constraints:

11) $0.00474U_1'+0.001466\ U_2'+0.09020U_3'-0.00020U_4'-0.0521U_5'\ 0.02530U_6' +$
$+ 0.01127U_7'-1.0U_{18} = 0.0712$

12) $0.00384U_1'-0.00061U_2'+0.05834U_3'+0.000272U_4'+0.00919U_5'+0.03004U_6' +$
$+ 0.02956U_7'-1.0U_{19} = 0.0656$

Upper bounds on design variables:

13) $U_1' + U_{20} = 1.20$

14) $U_2' + U_{21} = 1.20$

15) $U_3' + U_{22} = 1.20$

16) $U_4' + U_{23} = 1.20$

17) $U_5' + U_{24} = 1.20$

18) $U_6' + U_{25} = 1.20$

19) $U_7' + U_{26} = 1.20$

Lower bounds on design variables:

20) $U_1' - U_{27} = 0.80$

21) $U_2' - U_{28} = 0.80$

22) $U_3' - U_{29} = 0.80$

23) $U_4' - U_{30} = 0.80$

24) $U_5' - U_{31} = 0.80$

25) $U_6' - U_{32} = 0.80$

26) $U_7' - U_{33} = 0.80$

Minimum member areas:

27) $4.0\ U_1' - U_{34} = 2.12$

28) $5.0\ U_2' - U_{35} = 2.12$

29) $5.0\ U_3' - U_{36} = 2.12$

30) $5.0\ U_4' - U_{37} = 2.12$

31) $2.0\ U_5' - U_{38} = 2.12$

32) $4.0\ U_6' - U_{39} = 2.12$

33) $2.0\ U_7' - U_{40} = 2.12$

Objective function:

$$34)\quad 1028.0U'_1+168U'_2+218.U'_3+150.U'_4+150.U'_5+1028.U'_6 + \\ + 360.U'_7 = Z$$

Data cards were prepared (34 cards) and the optimization problem was solved directly by computer program LINPROG.

A reduction in area is indicated for all members except A_4 and A_5 which were a minimum originally. The total optimal weight of the truss is

$$Z = 2540.31 \text{ lbs}$$

as compared with preliminary design

$$W = 3082 \text{ lbs}$$

a saving in weight

$$\Delta W = 3082-2540.31 = 541.69 \text{ lbs}$$

$$\Delta W = \frac{541.69}{3082} \cdot 100 = 17.5\%$$

for one cycle. Not all of this saving can be realized as discrete sections cannot be matched exactly to the required area.

This result was accepted as a proof that the programs MINTWC and LINPROG are in working order and can be used for final design with the necessary changes to reduce the manual work on input.

Next, the same problem with h = 6'-0 was solved using the computer program MINTW as modified from MINTWC to compile all the necessary coefficients for objective function and constraints. See (4.2b). Then it was linked with LINPROG and DEMTR as subroutines. Member areas and allowable compressive stresses were taken from the preliminary design.

The optimum weight was reached in three cycles and again considerable saving was indicated.

Preliminary design weight	W_1	= 2296.37 lbs
Final design - optimal weight	W_4	= 1974.77 lbs
		321.60 lbs

$$\text{Saving } \frac{321.50}{2296.37} \times 100 = 14\%$$

5.2 DEM, MINTWC, LINPROG WITH MODIFICATIONS AS AN AUTOMATIC DESIGN SEQUENCE CHOOSING MEMBERS FROM A LIST

Preliminary design by DEMTDP.

The same design procedure is followed as that of Section 5.1 except that it is done automatically by computer.

1) By submitting data for a truss of depth 11'-0", the computer program DEMTDP designs 5 optimal weight trusses with one foot difference in depth between the successive

designs:

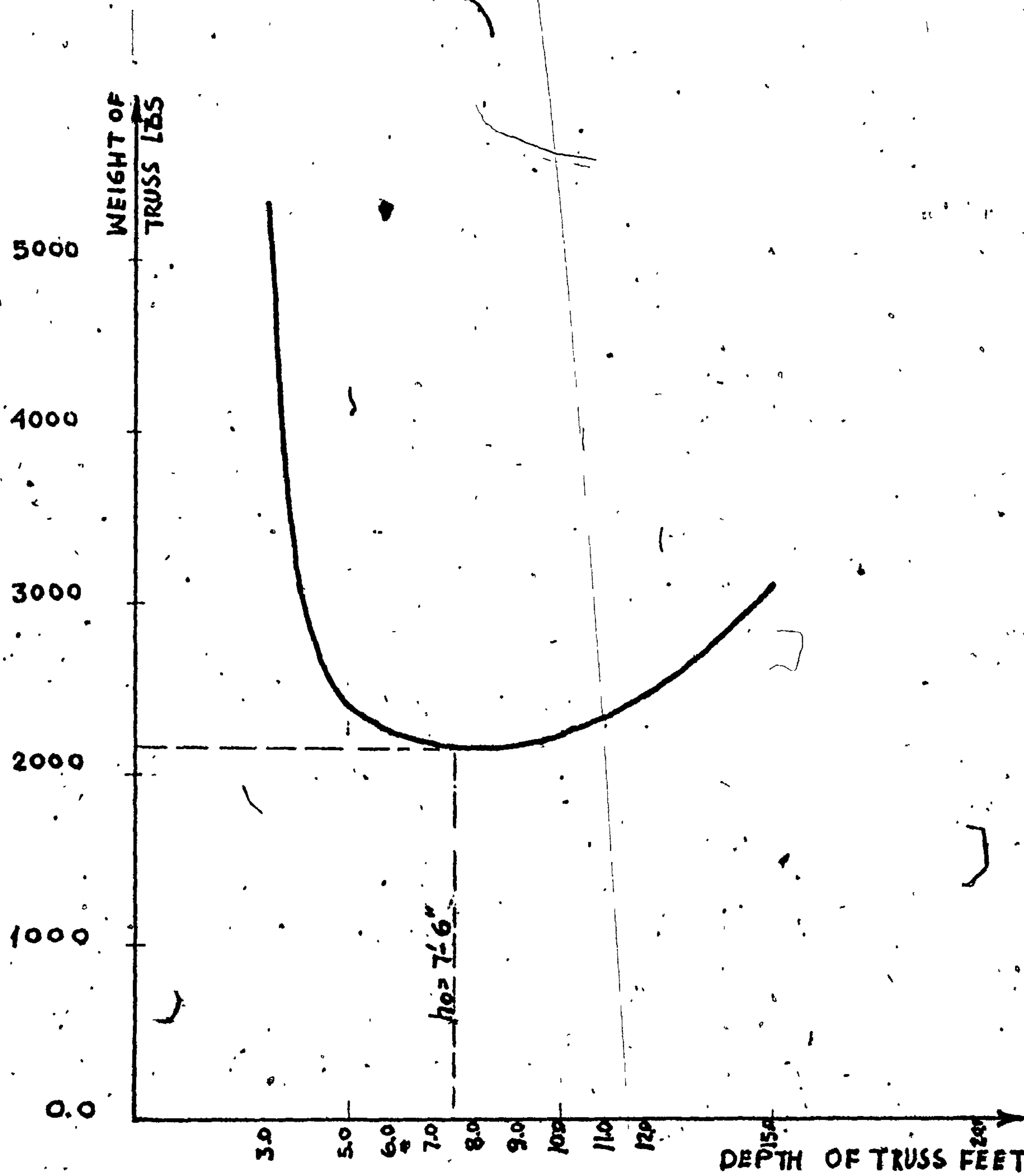
h = 11'	W = 2510.47 lbs
h = 10'	W = 2259.49 lbs
h = 9'	W = 2170.54 lbs
h = 8'	W = 2164.08 lbs
h = 7'	W = 2180.74 lbs

2) From data for a truss of depth h = 9':

h = 9'	W = 2169.50 lbs
h = 8'	W = 2164.08 lbs
h = 7'	W = 2180.74 lbs
h = 6'	W = 2278.99 lbs
h = 5'	W = 2288.52 lbs

3) From data for a truss of h = 15'-0"and step of 3'-0":

h = 15'	W = 3070.13 lbs
h = 12'	W = 2461.82 lbs
h = 9'	W = 2170.34 lbs
h = 6'	W = 2278.99 lbs
h = 3'	W = 5331.40 lbs



The Relationship Between the Depth of Truss and Weight of Truss Preliminary Analysis and Design by DEMTOP- Minimum Weight of Truss is at Approximately 7'-6"

GRAPH NO. 2

As we can see from overlapping designs the weights of the trusses are the same no matter what depth is initially submitted.

These results are displayed in Graph No. 2, on page 98, showing the relationship between the depth of the truss and the weight. The results are not quite identical to those of procedure 5.1, but they are close and indicate that the curve in the optimal area is quite flat. In this last procedure where we have more data in the optimal depth region, the optimal depth established by inspection is approximately 7'-6".

Final Design

In a normal design there will not be so many truss depths investigated and chances are remote that the optimal depth will coincide with one of the computed values. Therefore, the final design sequence is provided with a preliminary design part DMTD which will do preliminary design and optimization of the truss of the optimal depth the same way as DEMTDP. The data from there will be taken by MINTW, LINPROG and DEMTR which will do the final design and optimization by solving the linear programming problem.

The final design requires only one data submission of optimal depth h = 7'-6" and the member areas from the preliminary design of the truss with the depth closest to the optimal.

The final result was the truss of weight W = 2134.74 lbs and member sizes as follows:

Member 1	Section	2 ⌉⌈ 4 x 3 x 1/2	Area 6.50 in^2
Member 2	Section	2 ⌉⌈ 3 x 2 x 5/16	Area 2.93 in^2
Member 3	Section	2 ⌉⌈ 4 x 3 x 1/2	Area 6.50 in^2
Member 4	Section	2 ⌉⌈ $2\frac{1}{2}$ x $1\frac{1}{2}$ x 5/16	Area 2.30 in^2
Member 5	Section	2 ⌉⌈ $2\frac{1}{2}$ x $1\frac{1}{2}$ x 5/16	Area 2.30 in^2
Member 6	Section	2 ⌉⌈ $3\frac{1}{2}$ x 3 x 5/16	Area 3.86 in^2
Member 7	Section	2 ⌉⌈ $2\frac{1}{2}$ x 2 x 1/4	Area 2.12 in^2
Member 8	Section	2 ⌉⌈ 4 x 3 x 7/16	Area 5.74 in^2
Member 9	Section	2 ⌉⌈ 4 x 3 x 7/16	Area 5.74 in^2
Member 10	Section	2 ⌉⌈ $3\frac{1}{2}$ x 3 x 5/16	Area 3.86 in^2

In procedure 5.1 a truss of the depth h = 6'-0" was optimized with the total weight W = 1974.77 lbs. Here, it was done assuming that members of any area are available and that allowable compression stress remains constant during the design process.

The same truss with a depth h = 6'-0" was then optimized and designed by procedure 5.2 with members selected from a list. The optimal weight of the truss in this case, is larger than in procedure 5.2.

W = 2278.89 lbs

Such a result was to be expected and proves that for practical design procedure 5.2 is to be recommended.

CHAPTER 6

CONCLUSIONS

CHAPTER 6

CONCLUSIONS

(1) A design and optimization sequence added to a structural analysis program saves a considerable amount of material and designer's time.

(2) It is preferable to use discrete sections in the optimization process using a new calculated allowable buckling stress for each cycle.

(3) If the choice of depth of the truss is to be made by the structural designer, the proposed method of preliminary design with various depths is practical and can save more material than the final optimization of truss members for a given depth.

(4) Preliminary design to achieve fully stressed design uses comparatively little computer time and brings the design close to the optimal. It can also be used as the final design procedure where conditions do not warrant the expense of additional computer time for the final design and optimization by linear programming.

(5) For multiple loading conditions, and stress and deformation constraints, design and optimization by linear programming uses a considerable amount of computer time when the solution is sought by solving the primal problem. Savings in computer time can be effected by solving dual problems instead. This would require considerable modification of the MINTW and LINPROG computer programmes and was considered to be beyond the scope of this study.

REFERENCES

REFERENCES

[1] C.K. Wang, "Matrix Methods of Structural Analysis", International Textbook Co., Second Edition, 1970.

[2] J.M. Gere and W. Weaver, Jr., "Analysis of Framed Structures", D. Van Nostrand Co., 1963.

[3] T. Au and T.E. Stelson, "Introduction to Systems Engineering, Deterministic Models", Addison Wesley Publishing Co., 1969.

[4] G. Hadley, "Linear Programming", Addison-Wesley, Inc., 1961.

[5] G.B. Dantzig, "Linear Programming and Extensions", Princeton University Press, Princeton, New Jersey, 1963.

[6] S.I. Gass, "Linear Programming Methods and Applications", McGraw-Hill Co., 1969.

[7] A.M. Glicksman, "An Introduction to Linear Programming and the Theory of Games", Wiley & Sons, Inc., 1963.

[8] K.M. Romstad and C.K. Wang, "Optimum Design of Framed Structures", J. Struc. Div., ASCE, Dec. 1968, pp.2817-2845.

[9] F. Moses, "Optimum Structural Design Using Linear Programming", J. Struc. Div. ASCE, Dec. 1964, pp.89-104.

[10] D.M. Brown and A.H.S. Ang, "Structural Optimization by Nonlinear Programming", J. Struc. Div. ASCE, Dec.1966, pp.319-340.

[11] K.F. Reinschmitt, C.A. Cornell, J.F. Brotchie, "Iterative Design and Structural Optimization", J. Struc. Div. ASCE, Dec. 1966, pp.281-318.

[12] D. Johnson and D.M. Brotton, "Optimum Elastic Design of Redundant Trusses", J. Struc. Div., ASCE, Dec. 1969, pp.2589-2609.

[13] A.R. Toakley, "Optimum Design Using Available Sections", J. Struc. Div., ASCE, May 1968, pp.1219-1241.

[14] M. Save and W. Prager, "Minimum Weight Design of Beams Subject to Fixed and Moving Loads", J. Mech. Phys.Solids, Vol. 11, Pergamon Press, pp.255-267.

[15] M.F. Rubinstein and J. Karogasian,"Building Design Using Linear Programming", J. Struc. Div. ASCE, Dec. 1966, pp. 223-245.

[16] A. Cella and R.D. Logcher, "Automated Optimum Design From Discrete Components", J. Struc. Div. ASCE, Jan. 1971, pp. 175-190.

[17] K.F. Reinschmidt, "Discrete Structural Optimization", J. Struc. Div. ASCE, Jan. 1971, pp.133-156.

[18] L.A. Hill, Jr., "Automated Optimum Cost Building Design", J. Struc. Div. ASCE, Jan. 1966, pp.247-263.

[19] S.L. Lipson and A.D. Russell, "Cost Optimization of Structural Roof Systems", J. Struc. Div. ASCE, Aug. 1971, pp.2057-2071.

[20] T.P. Kicher, "Optimum Design-Minimum Weight Versus Fully Stressed", J. Struc. Div. ASCE, Dec. 1966, pp.265-279.

[21] D.F. Brooks and D.M. Brotton, "Computer System for Analysis of Large Frameworks", J. Struc. Div. ASCE, Dec. 1964, pp.1-23.

[22] N.M. Singaraj and J.K. Sridhar Rao, "Nonlinear Elastic Truss Analysis by Partitioning", J. Struc. Div., ASCE, Nov. 1972, pp.2399-2411.

[23] R.A. Ridha and R.N. Wright, "Minimum Weight Design of Frames", J. Struc. Div. ASCE, Aug. 1967, pp.165-183.

[24] P.M. Wright and C.E. Feng, "Optimum Design of Plane Frames Using a Multi-Mode Scheme", J.Eng. Inst. of Canada, Vol. 14, No. A-5, April 1971, pp.I-VI.

[25] D.F. Brooks and D.M. Brotton, "Formulation of a General Space Frame Program for a Large Computer", The Structural Engineer, Nov. 1966, No. 11, Vol. 44, pp.381-396.

[26] F. Moses, J.D. Stevenson, "Reliability-Based Structural Design", J. Struc. Div., ASCE, Feb. 1970, pp.221-244.

[27] J. Foulkes, "Minimum Weight Design and the Theory of Plastic Collapse", Quarterly Applied Mathematics, 1952, Vol. X, No. 4, pp.347-358.

[28] P. Pedersen, "Optimal Joint Positions for Space Trusses", J. Struc. Div. ASCE, Dec. 1973, pp.2459-2471.

[29] Behrooz Farshi, Lucien A.Schmidt,Jr.,"Minimum Weight Design of Stress Limited Trusses", J.Struc. Div. ASCE, Jan. 1974, pp.97-107.

APPENDIX A

DETAILS AND LISTING* OF COMPUTER PROGRAMS DEMTDP AND DEMTD

* Computer outputs used for the practical problem solved in this study (2 span truss) are available from the author.

APPENDIX A

DETAILS AND LISTING OF COMPUTER PROGRAMS DEMTDP AND DEMTD

A.1 DETAILS OF COMPUTER PROGRAM DEMTDP

This is basically the computer program "Direct Element Methods of Truss Analysis" [1]. It has been modified and extended to do preliminary design for a series of trusses with varying depth.

It analyzes a truss with given geometry, member sizes and loading. Then designs new members for maximum or minimum member forces according to C.S.A. S16 choosing members from a list, and calculates the weight of the truss. Then analysis is repeated with the new member areas, for new member forces new sections are chosen from the list, the weight of the truss is computed and compared with the weight from the previous design. If the difference between the two successive designs is more than 1% of the total truss weight, the design cycle is repeated until the difference is less than 1% or the maximum number of cycles is reached. For each cycle the deflections of preselected joints are checked and if they are larger than allowable all member areas are upgraded so that this violation is removed. This is then adopted as the optimum design for that depth of truss.

Then automatically, the depth of the truss is reduced by 12" and the same design procedure is repeated. At present, the program will design 5 trusses in a row with a depth varying by 12 inches between successive designs.

In the program: NP = degrees of freedom of truss
NF = number of members of truss
NLC = number of loading conditions

Max NP = 21
Max NF = 21
Max NLC = 10

The procedure of analysis is described in detail in Section 4.1. Some explanatory notes about the programs follow.

Lines 1-6	Dimension statements
7-23	Read and print design constants and list of available sections
24-28	Read and print member data
29-34	Computes constants used in designing members according to C.S.A. S16
35-140	Computes and prints X matrix
60	Computes the weight of the truss
83-84	Prints weight of truss
85-92	Check the difference of weight between two consecutive designs and also check if the maximum number of cycles is not exceeded

141-159	Computes and prints ratio X allow/X and selects the largest
160-182	Computes and prints member forces in order of loading conditions
183-196	Find maximum and minimum of member forces
197-245	Design truss members for maximum and minimum forces according to C.S.A. S16 choosing members from the list
246-275	Make areas of identical members equal
276	Repeat design cycle with new areas until difference in weight between the two successive cycles is less than 1% or when maximum number of cycles is reached. If deflection constraints are violated, member areas are adjusted upwards, final analysis with those areas is made and new members chosen to suit. This is then accepted as final design for that depth of truss.
291-297	Depth of truss is decreased by 12" and analysis and design is repeated for the new depth of truss.
298	END of program.

A.2 DETAILS OF COMPUTER PROGRAM DEMTD

This program consists of the main program DEMTD and three subroutines MINTW, LINPROG and DEMTR.

DEMTD and DEMTR are modifications of the computer program "Direct Element Method of Truss Analysis" [1] which is almost identical to DEMTDP except that the depth of the truss is not varied.

DEMTD designs and optimizes a truss of optimal depth as established in the preliminary design by the program DEMTDP.

The main program DEMTD repeats the same analysis and design sequence as DEMTDP but for one depth only.

Then the subroutine MINTW is called and computes the necessary coefficients for the linear programming problem.

Next is called subroutine LINPROG which solves the linear programming problems by the Simplex method and computes the optimal member areas.

Then subroutine DEMTR is called. It analyzes the truss with the optimal member areas, adjusts all member areas to remove deflection violations if any, and designs new members for the new member forces. The new member areas and new allowable stresses are returned to MINTW and the optimization process is repeated until the difference in weights between two successive designs is less than 0.5% or max. number of cycles has been completed. This is then taken as optimal design. Input and constants for DEMTD are the same as for DEMTDP and will not be repeated. Some explanatory notes in the programming components follow.

A2.1 Main Program

DEMTD is as DEMTDP except that the design is done for one depth only. (Optimal depth from graph of preliminary design).

A2.2 Subroutine MINTW

Lines 1-15	Base, dimensions and notations
16-30	Read and print additional data
31-283	Compute changes in member forces and joint deformations due to successive changes in member areas (increased 100%) one at a time
284-297	Generate slack variables for stress constraints
298-321	Generate coefficients for deformation constraints
322-332	Generate coefficients on diagonal
333-342	Make all stress constraint equations right-hand side positive.
343-357	Generate slack variables for deformation constraints

358-368	Make all deformation constraints right-hand side positive
369-390	Generate upper and lower bounds of design variables and constraints of minimum area for members
391-400	Generate coefficients for identical area equations
401-402	Print coefficients for Simplex tableau
403	Calls LINPROG
404	Calls DEMTR
407-416	Compare weights of the last two consecutive designs. If difference is more than 0.5% repeat the design cycle MINTW-LINPROG-DEMTR until it is less than 0.5% or the maximum number of cycles is completed whichever comes first. Return to DEMTD and end the program.

A2.3 Subroutine LINPROG

LINPROG solves by the Simplex method the standard linear programming problem given with all the coefficients for the tableau. The program adds artificial variables and solves the minimization problem.

Lines 1-11	Give basis, dimensions, establish constants for simplex tableau.
12-17	Generate artificial variables
18-22	Generate infeasibility functions.
23-27	Generate and prints column numbers for initial base
28-41	Select pivot column finding largest (absolute value) negative coefficients in "W" row
42-48	Check if B column is not negative. If negative print "infeasible solution" and stop program.
49-68	Generate ratios R(I) = A(I,NB)/A(I,IS). Find smallest R(I). If all A(I,IS) negative, print "Function unbounded" and stop program.
69-83	Solve for the new basis
84-98	Repeat cycle of solving infeasibility function until all coefficients on left-hand side in infeasibility function are positive or zero. If B column in infeasibility function is not zero print "infeasible solution" and stop program
94	Set MW = M + 1 and do optimization cycles for objective function
99-106	Print final tableau of optimum solution

	and values of variables
107-116	Print all design variables in ascending order
117	Compute the new optimal member areas and print
123	Return
124	END

A2.4 Subroutine DEMTR

This program is almost identical to DEMTD. It analyses the structure with the new optimal member areas from LINPROG, upgrades all areas if some deflection constraints are violated and designs new members for the new member forces. At the end, the program computes the truss weight and returns to MINTW for a new cycle if the difference in truss weights between two consecutive designs is more than 0.5%.

A.3 LISTING OF COMPUTER PROGRAM DEMTDP

```
      PROGRAM DEMTOP(INPUT,OUTPUT)
      DIMENSION ASAT(22,22),P(21,8),X(22,8),F(10),INDEX(21)
      DIMENSION MNO(21),NPN(21,4),HVA(21,3)
      DIMENSION      FALLOW(22),W(10),RX(22,8)
      DIMENSION NA(60,5),NS(28),FMIN(28),FMAX(28),NO(60),NAME(60,2)
      REAL NA
      READ 101,NP,NF,NLC,KCM,KHM,NL,NLMIN,NTP1,NTP2
  101 FORMAT (9I5)
      PRINT 101,NP,NF,NLC,KCM,KHM,NL,NLMIN,NTP1,NTP2
      READ 102, E,FY,XALL
  102 FORMAT(2X,3F10.4)
      PRINT 102, E,FY,XALL
      PRINT 252
 252  FORMAT (25X,17HLIST OF SECTIONS.)
      DO 250 I=1,NL
 250  READ 251,NO(I),(NAME(I,J),J=1,2),(NA(I,J),J=1,5)
 251  FORMAT (I5,2X,A10,A6,5F10.4)
      DO 253 I=1,NL
 253  PRINT254,NO(I),(NAME(I,J),J=1,2),(NA(I,J),J=1,5)
 254  FORMAT (I5,2X,A10,A6,5F10.4)
C     NO(I)= IDENTIFICATION NUMBER.
C     NA(I,1)= AREA  NA(I,2)= RX  NA(I,3)= RY
C     NA(I,4)= B WIDTH OF FLANGE.  NA(I,5)= T THICKNESS OF FLANGE.
      PRINT 104
  104 FORMAT(40H0DIRECT ELEMENT METHOD OF TRUSS ANALYSIS//)
      DO 206 I=1,NF
 206  READ 106,MNO(I),(NPN(I,J),J=1,4),(HVA(I,J),J=1,3),FALLOW(I)
 106  FORMAT(5I5,4F10.4)
      FALLT=0.6*FY
      CO=30.0-FY/5.0
      IF(CO.GT.20.0) CO=20.0
      CP=SQRT(286000.0/(FY-13.0))
      IF(CP.LT.78.0) CP=78.0
      XM=(0.60*FY-149000.0/(CP*CP))/(CP-CO)
      L3=NTP1
      L6=NTP2
      KH=1
      KC=1
 204  NPP1=NP+1
      DO 103 I=1,NPP1
      DO 103 J=1,NPP1
  103 ASAT(I,J)=0.
      PRINT 105
  105 FORMAT (27H0MEMBER   NP1   NP2   NP3   NP4, 7X, 1HH, 11X, 1HV, 11X,
     1  1HA, 11X, 1HL, 11X, 3HCOS, 12X, 3HSIN,12X,6HFALLOW,12X,2HNS/)
 205  J=KC
      W(J)=0.0
      DO 108 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
```

```
          A=HVA(I,3)
          XL=SQRT(H*H+V*V)
          XCOS=H/XL
          XSIN=V/XL
          W(J)=W(J)+A*XL*0.284
          PRINT 107,MEMNO,NP1,NP2,NP3,NP4,H,V,A,XL,XCOS,XSIN,FALLOW(I)
         1,NS(I)
  107     FORMAT (1H0,I5,I6,3I5,4F12.4,3F15.8,5X,I5)
          TEMP1=E*A*XCOS*XCOS/XL
          TEMP2=E*A*XCOS*XSIN/XL
          TEMP3=E*A*XSIN*XSIN/XL
          ASAT(NP1,NP1)=ASAT(NP1,NP1)+TEMP1
          ASAT(NP1,NP2)=ASAT(NP1,NP2)+TEMP2
          ASAT(NP1,NP3)=ASAT(NP1,NP3)-TEMP1
          ASAT(NP1,NP4)=ASAT(NP1,NP4)-TEMP2
          ASAT(NP2,NP1)=ASAT(NP2,NP1)+TEMP2
          ASAT(NP2,NP2)=ASAT(NP2,NP2)+TEMP3
          ASAT(NP2,NP3)=ASAT(NP2,NP3)-TEMP2
          ASAT(NP2,NP4)=ASAT(NP2,NP4)-TEMP3
          ASAT(NP3,NP1)=ASAT(NP3,NP1)-TEMP1
          ASAT(NP3,NP2)=ASAT(NP3,NP2)-TEMP2
          ASAT(NP3,NP3)=ASAT(NP3,NP3)+TEMP1
          ASAT(NP3,NP4)=ASAT(NP3,NP4)+TEMP2
          ASAT(NP4,NP1)=ASAT(NP4,NP1)-TEMP2
          ASAT(NP4,NP2)=ASAT(NP4,NP2)-TEMP3
          ASAT(NP4,NP3)=ASAT(NP4,NP3)+TEMP2
  108     ASAT(NP4,NP4)=ASAT(NP4,NP4)+TEMP3
          PRINT 140,J,W(J)
  140     FORMAT (25X,7HWEIGHT ,I2,3H = ,F10.4)
          IF(KC.EQ.KCM-1) GO TO 215
          J=KC
          IF(J.EQ.1) GO TO 215
          IF(J.GT.1) J1=J-1
          T=ABS(W(J1)-W(J))/W(J1)
          IF(T.LT.0.010)GO TO 138
          IF (KC.GE.KCM) GO TO 138
  215     DO 110 I=1,NP
  110     INDEX(I)=0
  111     AMAX=-1.
          DO 114 I=1,NP
          IF (INDEX(I)) 114,112,114
  112     TEMP=ABS(ASAT(I,I))
          IF (TEMP-AMAX) 114,114,113
  113     ICOL=I
          AMAX=TEMP
  114     CONTINUE
          IF (AMAX) 120,138,115
  115     INDEX (ICOL)=1
          PIVOT=ASAT(ICOL,ICOL)
          ASAT(ICOL,ICOL)=1.0
          PIVOT=1./PIVOT
          DO 116 J=1,NP
  116     ASAT(ICOL,J)=ASAT(ICOL,J)*PIVOT
          DO 119 I=1,NP
          IF (I-ICOL) 117,119,117
```

```
  117 TEMP=ASAT(I,ICOL)
      ASAT(I,ICOL)=0.0
      DO 118 J=1,NP
  118 ASAT(I,J)=ASAT(I,J)-ASAT(ICOL,J)*TEMP
  119 CONTINUE
      GO TO 111
  120 CONTINUE
      IF(KC.GT.1) GO TO 125
      DO 121 I=1,NP
      DO 121 J=1,NLC
  121 P(I,J)=0.
  122 READ 123,I,J,PIJ
  123 FORMAT (2I5,F10.4)
      IF (I) 125,125,124
  124 P(I,J)=PIJ
      GO TO 122
  125 PRINT 126
  126 FORMAT (13H0THE MATRIX P)
      DO 127 I=1,NP
  127 PRINT 128,I,(P(I,J),J=1,NLC)
  128 FORMAT (4H ROW,I3,1X,4(1PE16.7)/(8X,4(1PE16.7)))
      DO 129 I=1,NP
      DO 129 J=1,NLC
      X(I,J)=0.
      DO 129 K=1,NP
  129 X(I,J)=X(I,J)+ASAT(I,K)*P(K,J)
      PRINT 130
  130 FORMAT (13H0THE MATRIX X)
      DO 131 I=1,NP
  131 PRINT 128,I,(X(I,J),J=1,NLC)
      DO 201 I=1,NP
      DO 200 J=1,NLC
      IF(X(I,J)) 260,261,261
  260 RX(I,J)=X(I,J)/(-XALL)
      GO TO 200
  261 RX(I,J)=X(I,J)/XALL
  200 CONTINUE
  201 CONTINUE
      DO 210 I=L3,L6,3
      PRINT 211,(RX(I,J),J=1,NLC)
  211 FORMAT(33X,F10.4)
  210 CONTINUE
      RXMAX=RX(L3,1)
      DO 302 I=L3,L6,3
      DO 302 J=1,NLC
      IF(RXMAX.LT.RX(I,J)) RXMAX=RX(I,J)
  302 CONTINUE
      PRINT 310,RXMAX
  310 FORMAT(25X,8HRXMAX = F10.4)
      PRINT 132
  132 FORMAT (49H1THE MEMBER FORCES IN ORDER OF LOADING CONDITIONS/)
      DO 133 J=1,NLC
  133 X(NPP1,J)=0.
      DO 134 I=1,NF
      MEMNO=MNO(I)
```

```
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      XCOS=H/XL
      XSIN=V/XL
      FMIN(I)=0.0
      FMAX(I)=0.0
      DO 136 J=1,NLC
      F(J)=E*A/XL*(XCOS*(X(NP3,J)-X(NP1,J))+XSIN*(X(NP4,J)-X(NP2,J)))
  136 CONTINUE
      PRINT 137, MEMNO,(F(J),J=1,NLC)
  137 FORMAT (1H ,I5,5F12.4/(6X,5F12.4))
C
C     FIND MAX AND MIN MEMBER FORCES.
C
      DO 265 J=1,NLC
      IF(F(J)) 202,203,203
  202 IF (FMIN(I).GT.F(J)) FMIN(I)=F(J)
      GO TO 265
  203 IF(FMAX(I).LT.F(J)) FMAX(I) =F(J)
  265 CONTINUE
      PRINT 266 ,MEMNO   ,FMIN(I),FMAX(I),XL
  266 FORMAT(1H ,I5,3F12.4)
C
C     FIND SMALLEST SECTION FOR TENSION.
C
      AT=FMAX(I)/FALLT
      I1=NLMIN
      IF (ABS(AT).LT.0.0001) GO TO 236
  235 IF(NA(I1,1).LT.AT) GO TO 234
      IF(KC.GE.KCM-1) GO TO 404
      GO TO 405
  404 IF(NA(I1,1).LT.HVA(I,3)*RXMAX)GO TO 234
  405 IF(NA(I1,2).LE.NA(I1,3))RI1=NA(I1,2)
      IF(NA(I1,2).GT.NA(I1,3))RI1=NA(I1,3)
      IF   (RI1.LT.XL/300.0) GO TO 234
      IF (ABS(FMIN(I)).LT.0.0001) GO TO 270
      GO TO 236
  234 I1=I1+1
      IF(I1.GT.NL) GO TO 240
      GO TO 235
C
C     FIND SMALLEST SECTION FOR COMPRESSION AND TEST FOR LOCAL BUCKLING.
C
  236 I2=I1
  237 IF(NA(I2,2).LE.NA(I2,3))RI2=NA(I2,2)
      IF(NA(I2,2).GT.NA(I2,3))RI2= NA(I2,3)
      IF   (RI2.LT.XL/200.0) GO TO 238
      FY=44.00
      B=NA(I2,4)
```

```
      T=NA(I2,5)
      IF(B/T.GT.(75.0/SQRT(FY)))FY=1.67*(FY-8.6)-0.067*(FY-21.6)*B/T
C
C     COMPUTE ALLOW. COMPRESSION STRESS FOR EVERY SECTION UNTIL A SATISFACTO
C     ONE IS FOUND.
      IF(XL/RI2.LE.CO) FAI2=0.60*FY
      IF(XL/RI2.GT.CO.AND.XL/RI2.LE.CP) FAI2=0.60*FY-
     1XM*(XL/RI2-CO)
      IF(XL/RI2.GT.CP) FAI2=149000.0/((XL/RI2)*(XL/RI2))
      IF(ABS(FAI2*NA(I2,1)) .LT.ABS(FMIN(I))) GO TO 238
      IF(KC.GE.KCM-1) GO TO 406
      GO TO 239
  406 IF(NA(I2,1).LT.HVA(I,3)*RXMAX)GO TO 238
      GO TO 239
  238 I2=I2+1
      IF(I2.GT.NL) GO TO 240
      GO TO 237
  270 NS(I)=I1
      HVA(I,3)=NA(I1,1)
      FALLOW(I)=-0.1
      GO TO 134
  239 NS(I)=I2
      HVA(I,3)=NA(I2,1)
      FALLOW(I)=-FAI2
  134 CONTINUE
      IF (HVA(1,3).LT.HVA(3,3)) GO.TO 300
      IF(HVA(1,3).GE.HVA(3,3)) GO TO 301
  307 IF (HVA(4,3).LT.HVA(5,3))  GO TO 332
      IF (HVA(4,3).GE.HVA(5,3)) GO TO 303
  308 IF (HVA(8,3).LT.HVA(9,3))  GO TO 304
      IF(HVA(8,3).GE.HVA(9,3))   GO TO 305
      GO TO 309
  300 HVA(1,3)=HVA(3,3)
      NS(1)=NS(3)
      FALLOW(1)=FALLOW(3)
      GO TO 307
  301 HVA(3,3)=HVA(1,3)
      NS(3)=NS(1)
      FALLOW(3)=FALLOW(1)
      GO TO 307
  332 HVA(4,3)=HVA(5,3)
      NS(4)=NS(5)
      FALLOW(4)=FALLOW(5)
      GO TO 308
  303 HVA(5,3)=HVA(4,3)
      NS(5)=NS(4)
      FALLOW(5)=FALLOW(4)
      GO TO 308
  304 HVA(8,3)=HVA(9,3)
      NS(8)=NS(9)
      FALLOW(8)=FALLOW(9)
      GO TO 309
  305 HVA(9,3)=HVA(8,3)
      NS(9)=NS(8)
      FALLOW(9)=FALLOW(8)
```

```
      309 CONTINUE
          KC=KC+1
          GO TO 204
      138 IF(KC.EQ.KCM) GO TO 139
          KC=KCM-1
          IF(RXMAX.GT.1.0) GO TO 407
          GO TO 139
      407 DO 408 I=1,NF
      408 HVA(I,3)=HVA(I,3)*RXMAX
          GO TO 204
      240 PRINT 241
      241 FORMAT(41H0INFEASIBLE WITH PRESENT LIST OF SECTIONS)
      139 KH=KH+1
          IF (KH.GT.KHM) GO TO 410
          KC=2
          DO 350 I=1,NF
          IF(HVA(I,2).LT.0.0) HVA(I,2)=HVA(I,2)+12.0
          IF(HVA(I,2).GT.0.0) HVA(I,2)=HVA(I,2)-12.0
          IF(HVA(I,2).EQ.0.0) HVA(I,2)=HVA(I,2)+0.00
      350 CONTINUE
          GO TO 204
      410 STOP
          END
MORE MEMORY WOULD HAVE RESULTED IN BETTER OPTIMIZATION
```

A.4 LISTING OF COMPUTER PROGRAM DEMTD WITH SUBROUTINES MINTW, LINPROG, DEMTR

A.4.1 LISTING OF DEMTD

```
      PROGRAM DEMTD(INPUT=1018,OUTPUT=1018)
      DIMENSION ASAT(22,22),P(21,8),X(22,8),F(10),INDEX(21)
      DIMENSION MNO(21),NPN(21,4),HVA(21,3)
      DIMENSION       FALLOW(22),W(10),RX(22,8)
      DIMENSION NA(60,5),NS(28),FMIN(28),FMAX(28),NO(60),NAME(60,2)
      REAL NA
      READ 101,NP,NF,NLC,KCM,KHM,NL,NLMIN,NTP1,NTP2
  101 FORMAT (9I5)
      PRINT 101,NP,NF,NLC,KCM,KHM,NL,NLMIN,NTP1,NTP2
      READ 102, E,FY,XALL
  102 FORMAT(2X,3F10.4)
      PRINT 102, E,FY,XALL
      PRINT 252
  252 FORMAT (25X,17HLIST OF SECTIONS.)
      DO 250 I=1,NL
  250 READ 251,NO(I),(NAME(I,J),J=1,2),(NA(I,J),J=1,5)
  251 FORMAT (I5,2X,A10,A6,5F10.4)
      DO 253 I=1,NL
  253 PRINT254,NO(I),(NAME(I,J),J=1,2),(NA(I,J),J=1,5)
  254 FORMAT (I5,2X,A10,A6,5F10.4)
C     NO(I)= IDENTIFICATION NUMBER.
C     NA(I,1)= AREA  NA(I,2)= RX  NA(I,3)= RY
C     NA(I,4)= B WIDTH OF FLANGE.  NA(I,5)= T THICKNESS OF FLANGE.
      PRINT 104
  104 FORMAT(40H0DIRECT ELEMENT METHOD OF TRUSS ANALYSIS//)
      DO 206 I=1,NF
  206 READ 106,MNO(I),(NPN(I,J),J=1,4),(HVA(I,J),J=1,3),FALLOW(I)
  106 FORMAT(5I5,4F10.4)
      FALLT=0.6*FY
      CO=30.0-FY/5.0
      IF(CO.GT.20.0) CO=20.0
      CP=SQRT(286000.0/(FY-13.0))
      IF(CP.LT.78.0) CP=78.0
      XM=(0.60*FY-149000.0/(CP*CP))/(CP-CO)
      L3=NTP1
      L6=NTP2
      KH=1
      KC=1
  204 NPP1=NP+1
      DO 103 I=1,NPP1
      DO 103 J=1,NPP1
  103 ASAT(I,J)=0.
      PRINT 105
  105 FORMAT (27HOMEMBER   NP1  NP2  NP3  NP4, 7X, 1HH, 11X, 1HV, 11X,
     1  1HA, 11X, 1HL, 11X, 3HCOS, 12X, 3HSIN,12X,6HFALLOW,12X,2HNS/)
  205 J=KC
      W(J)=0.0
      DO 108 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
```

```
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      XCOS=H/XL
      XSIN=V/XL
      IF(KC.EQ.KCM-1) GO TO 411
      W(J)=W(J)+A*XL*0.284
      PRINT 107,MEMNO,NP1,NP2,NP3,NP4,H,V,A,XL,XCOS,XSIN,FALLOW(I)
     1,NS(I)
  107 FORMAT (1H0,I5,I6,3I5,4F12.4,3F15.8,5X,I5)
  411 TEMP1=E*A*XCOS*XCOS/XL
      TEMP2=E*A*XCOS*XSIN/XL
      TEMP3=E*A*XSIN*XSIN/XL
      ASAT(NP1,NP1)=ASAT(NP1,NP1)+TEMP1
      ASAT(NP1,NP2)=ASAT(NP1,NP2)+TEMP2
      ASAT(NP1,NP3)=ASAT(NP1,NP3)-TEMP1
      ASAT(NP1,NP4)=ASAT(NP1,NP4)-TEMP2
      ASAT(NP2,NP1)=ASAT(NP2,NP1)+TEMP2
      ASAT(NP2,NP2)=ASAT(NP2,NP2)+TEMP3
      ASAT(NP2,NP3)=ASAT(NP2,NP3)-TEMP2
      ASAT(NP2,NP4)=ASAT(NP2,NP4)-TEMP3
      ASAT(NP3,NP1)=ASAT(NP3,NP1)-TEMP1
      ASAT(NP3,NP2)=ASAT(NP3,NP2)-TEMP2
      ASAT(NP3,NP3)=ASAT(NP3,NP3)+TEMP1
      ASAT(NP3,NP4)=ASAT(NP3,NP4)+TEMP2
      ASAT(NP4,NP1)=ASAT(NP4,NP1)-TEMP2
      ASAT(NP4,NP2)=ASAT(NP4,NP2)-TEMP3
      ASAT(NP4,NP3)=ASAT(NP4,NP3)+TEMP2
  108 ASAT(NP4,NP4)=ASAT(NP4,NP4)+TEMP3
      IF(KC.EQ.KCM-1) GO TO 215
      PRINT 140,J,W(J)
  140 FORMAT (25X,7HWEIGHT ,I2,3H = ,F10.4)
      J=KC
      IF(J.EQ.1) GO TO 215
      IF(J.GT.1) J1=J-1
      T=ABS(W(J1)-W(J))/W(J1)
      IF(T.LT.0.010)GO TO 138
      IF (KC.GE.KCM) GO TO 138
  215 DO 110 I=1,NP
  110 INDEX(I)=0
  111 AMAX=-1.
      DO 114 I=1,NP
      IF (INDEX(I)) 114,112,114
  112 TEMP=ABS(ASAT(I,I))
      IF (TEMP-AMAX) 114,114,113
  113 ICOL=I
      AMAX=TEMP
  114 CONTINUE
      IF (AMAX) 120,138,115
  115 INDEX (ICOL)=1
      PIVOT=ASAT(ICOL,ICOL)
      ASAT(ICOL,ICOL)=1.0
      PIVOT=1./PIVOT
      DO 116 J=1,NP
  116 ASAT(ICOL,J)=ASAT(ICOL,J) *PIVOT
      DO 119 I=1,NP
```

```
      IF (I-ICOL) 117,119,117
  117 TEMP=ASAT(I,ICOL)
      ASAT(I,ICOL)=0.0
      DO 118 J=1,NP
  118 ASAT(I,J)=ASAT(I,J)-ASAT(ICOL,J)*TEMP
  119 CONTINUE
      GO TO 111
 120  CONTINUE
      IF(KC.GT.1) GO TO 125
      DO 121 I=1,NP
      DO 121 J=1,NLC
  121 P(I,J)=0.
  122 READ 123,I,J,PIJ
  123 FORMAT (2I5,F10.4)
      IF (I) 125,125,124
  124 P(I,J)=PIJ
      GO TO 122
  125 PRINT 126
  126 FORMAT (13H0THE MATRIX P)
      DO 127 I=1,NP
 127  PRINT 128,I,(P(I,J),J=1,NLC)
  128 FORMAT (4H ROW,I3,1X,4(1PE16.7)/(8X,4(1PE16.7)))
      DO 129 I=1,NP
      DO 129 J=1,NLC
      X(I,J)=0.
      DO 129 K=1,NP
  129 X(I,J)=X(I,J)+ASAT(I,K)*P(K,J)
      PRINT 130
  130 FORMAT (13H0THE MATRIX X)
      DO 131 I=1,NP
 131  PRINT 128,I,(X(I,J),J=1,NLC)
      DO 201 I=1,NP
      DO 200 J=1,NLC
      IF(X(I,J)) 260,261,261
 260  RX(I,J)=X(I,J)/(-XALL)
      GO TO 200
 261  RX(I,J)=X(I,J)/XALL
 200  CONTINUE
 201  CONTINUE
      DO 210 I=L3,L6,3
      PRINT 211,(RX(I,J),J=1,NLC)
 211  FORMAT(33X,F10.4)
 210  CONTINUE
      RXMAX=RX(L3,1)
      DO 302 I=L3,L6,3
      DO 302 J=1,NLC
      IF(RXMAX.LT.RX(I,J)) RXMAX=RX(I,J)
 302  CONTINUE
      PRINT 310,RXMAX
 310  FORMAT(25X,8HRXMAX = F10.4)
      PRINT 132
  132 FORMAT (49H1THE MEMBER FORCES IN ORDER OF LOADING CONDITIONS/)
      DO 133 J=1,NLC
 133  X(NPP1,J)=0.
      DO 134 I=1,NF
```

```
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      XCOS=H/XL
      XSIN=V/XL
      FMIN(I)=0.0
      FMAX(I)=0.0
      DO 136 J=1,NLC
      F(J)=E*A/XL*(XCOS*(X(NP3,J)-X(NP1,J))+XSIN*(X(NP4,J)-X(NP2,J)))
  136 CONTINUE
      PRINT 137, MEMNO,(F(J),J=1,NLC)
  137 FORMAT (1H ,I5,5F12.4/(6X,5F12.4))
C
C     FIND MAX AND MIN MEMBER FORCES.
C
      DO 265 J=1,NLC
      IF(F(J)) 202,203,203
  202 IF (FMIN(I).GT.F(J)) FMIN(I)=F(J)
      GO TO 265
  203 IF(FMAX(I).LT.F(J)) FMAX(I) =F(J)
  265 CONTINUE
      PRINT 266 ,MEMNO   ,FMIN(I),FMAX(I),XL
  266 FORMAT(1H ,I5,3F12.4)
C
C     FIND SMALLEST SECTION FOR TENSION.
C
      AT=FMAX(I)/FALLT
      I1=NLMIN
      IF (ABS(AT).LT.0.0001) GO TO 236
  235 IF(NA(I1,1).LT.AT) GO TO 234
      IF(KC.GE.KCM-1) GO TO 404
      GO TO 405
  404 IF(NA(I1,1).LT.HVA(I,3)*PXMAX)GO TO 234
  405 IF(NA(I1,2).LE.NA(I1,3))RI1=NA(I1,2)
      IF(NA(I1,2).GT.NA(I1,3))RI1=NA(I1,3)
      IF    (RI1.LT.XL/300.0) GO TO 234
      IF (ABS(FMIN(I)).LT.0.0001) GO TO 270
      GO TO 236
  234 I1=I1+1
      IF(I1.GT.NL) GO TO 240
      GO TO 235
C
C     FIND SMALLEST SECTION FOR COMPRESSION AND TEST FOR LOCAL BUCKLING.
C
  236 I2=I1
  237 IF(NA(I2,2).LE.NA(I2,3))RI2=NA(I2,2)
      IF(NA(I2,2).GT.NA(I2,3))RI2= NA(I2,3)
      IF    (RI2.LT.XL/200.0) GO TO 238
      FY=44.00
```

```
          B=NA(I2,4)
          T=NA(I2,5)
          IF(B/T.GT.(75.0/SQRT(FY)))FY=1.672*(FY-8.6)-0.067*(FY-21.6)*B/T
C
C         COMPUTE ALLOW. COMPRESSION STRESS FOR EVERY SECTION UNTIL A SATISFACTO
C         ONE IS FOUND.
          IF(XL/RI2.LE.CO) FAI2=0.60*FY
          IF(XL/RI2.GT.CO.AND.XL/RI2.LE.CP) FAI2=0.60*FY-
         1XM*(XL/RI2-CO)
          IF(XL/RI2.GT.CP) FAI2=149000.0/((XL/RI2)*(XL/RI2))
          IF(ABS(FAI2*NA(I2,1)) .LT.ABS(FMIN(I))) GO TO 238
          IF(KC.GE.KCM-1) GO TO 406
          GO TO 239
  406     IF(NA(I2,1).LT.HVA(I,3)*RXMAX)GO TO 238
          GO TO 239
  238     I2=I2+1
          IF(I2.GT.NL) GO TO 240
          GO TO 237
  270     NS(I)=I1
          HVA(I,3)=NA(I1,1)
          FALLOW(I)=-0.1
          GO TO 134
  239     NS(I)=I2
          HVA(I,3)=NA(I2,1)
          FALLOW(I)=-FAI2
  134     CONTINUE
          IF (HVA(1,3).LT.HVA(3,3)) GO TO 300
          IF(HVA(1,3).GE.HVA(3,3)) GO TO 301
  307     IF (HVA(4,3).LT.HVA(5,3))  GO TO 332
          IF (HVA(4,3).GE.HVA(5,3)) GO TO 303
  308     IF (HVA(8,3).LT.HVA(9,3))  GO TO 304
          IF(HVA(8,3).GE.HVA(9,3))  GO TO 305
          GO TO 309
  300     HVA(1,3)=HVA(3,3)
          NS(1)=NS(3)
          FALLOW(1)=FALLOW(3)
          GO TO 307
  301     HVA(3,3)=HVA(1,3)
          NS(3)=NS(1)
          FALLOW(3)=FALLOW(1)
          GO TO 307
  332     HVA(4,3)=HVA(5,3)
          NS(4)=NS(5)
          FALLOW(4)=FALLOW(5)
          GO TO 308
  303     HVA(5,3)=HVA(4,3)
          NS(5)=NS(4)
          FALLOW(5)=FALLOW(4)
          GO TO 308
  304     HVA(8,3)=HVA(9,3)
          NS(8)=NS(9)
          FALLOW(8)=FALLOW(9)
          GO TO 309
  305     HVA(9,3)=HVA(8,3)
          NS(9)=NS(8)
```

```
      FALLOW(9)=FALLOW(8)
  309 CONTINUE
      KC=KC+1
      GO TO 204
  138 IF(RXMAX.GT.1.00) GO TO 407
      GO TO 139
  407 KC=KCM-1
      DO 408 I=1,NF
  408 HVA(I,3)=HVA(I,3)*RXMAX
      GO TO 204
  240 PRINT 241
  241 FORMAT(41H0INFEASIBLE WITH PRESENT LIST OF SECTIONS)
      GO TO 410
  139 CALL MINTW (NP,NF,NLC,E,MNO,NPN,HVA,FALLOW,FALLT,P,KH,
     1NQ,NAME,NA,NS,FMIN,FMAX,W,CO,CP,XM,KC,KHM,
     1NL,NLMIN,FY,KCM)
  410 STOP
      END
```

A4.2 LISTING OF SUBROUTINE MINTW

```
      SUBROUTINE MINTW (NTP,NF,NLC,E,MNO,NPN,HVA,FALLOW,FALLT,P,KH,
     1NO,NAME,NA,NS,FMIN,FMAX,W,CO,CP,XM,KC,KHM,
     1NL,NLMIN,FY,KCM)
      DIMENSION ASAT(22,22),P(21,8),X(22,8),F(10),INDEX(21)
      DIMENSION NP(4),SA(4),SAK(22,22),DELF(10),NPN(21,4),HVA(21,3)
      DIMENSION ASB(22,22),DELX(8),C(22,22),MNO(21)
      DIMENSION BASE(85,175),FALLOW(22)
      DIMENSION NA(60,5),NS(28),FMIN(28),FMAX(28),NO(60),NAME(60,2)
      DIMENSION W(10)
      REAL NA
C     NTP=TOTAL NO OF FREEDOMS. NF= TOTAL NO OF MEMBERS. NLC= TOTAL NOOF
C     LOADING CONDITIONS NID= NO OF IDENYITY EQ. NCP= TOTAL NO CONSTRAINED
C     DEFORMATIONS NTP1, NTP2 NUMBFRS OF CONSTRAINTED FREEDOMS.UB, XLB=UPPER
C     LOWER BOUNDS ON DESIGN VARIABLES. AMIN=MINIMUM AREAFOR A MEMBER
C     XALL=ALLOWABLE DEFLECTION.
      READ 101,NID,NCP,NTP1,NTP2
  101 FORMAT (16I5)
      READ 102,UB,XLB,AMIN,XALL
  102 FORMAT(8F10.4)
      LL1=NF+2*(NF*NLC+NCP*NLC+3*NF) +NID+1
      LL2=-NF+NF*NLC+NCP*NLC+3*NF
      M1=NF*NLC+NCP*NLC+NF*3+NID+1
      M3=M1-1
   10 PRINT 800
  800 FORMAT (1X,3HNTP,2X,2HNF,3X,3HNLC,2X,3HNID,2X,3HNCP,2X,4HNTP1,1X,
     1 4HNTP2/)
      PRINT 101 , NTP,NF,NLC,NID,NCP,NTP1,NTP2
  811 PRINT 810
  810 FORMAT (3X,1HE,9X,2HUB,8X,3HHLB,7X,4HAMIN,6X,4HXALL,5X,5HFALLT/)
      PRINT 102,E,UB,XLB,AMIN,XALL,FALLT
  820 NPP1=NTP+1
      DO 301 I=1,M1
      DO 301 J=1,LL1
  301 BASE (I,J)=0.0
      DO 103 I=1,NPP1
      DO 103 J=1,NPP1
  103 ASAT(I,J)=0.
  812 PRINT 105
  105 FORMAT (27H0MEMBER  NP1  NP2  NP3  NP4, 7X, 1HH, 11X, 1HV, 11X,
     I  1HA, 11X, 1HL, 11X, 3HCOS, 12X, 3HSIN,12X,6HFALLOW/)
      DO 11 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
C
C     CALCULATE ORIGINAL WEIGHT OF TRUSS.
C
      BASE(M1,I)=A*XL*0.284
      CS=H/XL
```

```
      SN=V/XL
      PRINT 107,MEMNO,NP1,NP2,NP3,NP4,H,V,A,XL,CS,SN,FALLOW(I)
  107 FORMAT     (I5,I6,3I5,4F12.4,3F15.8)
      TEMP1=E*A*CS*CS/XL
      TEMP2=E*A*CS*SN/XL
      TEMP3=E*A*SN*SN/XL
      ASAT(NP1,NP1)=ASAT(NP1,NP1)+TEMP1
      ASAT(NP1,NP2)=ASAT(NP1,NP2)+TEMP2
      ASAT(NP1,NP3)=ASAT(NP1,NP3)-TEMP1
      ASAT(NP1,NP4)=ASAT(NP1,NP4)-TEMP2
      ASAT(NP2,NP1)=ASAT(NP2,NP1)+TEMP2
      ASAT(NP2,NP2)=ASAT(NP2,NP2)+TEMP3
      ASAT(NP2,NP3)=ASAT(NP2,NP3)-TEMP2
      ASAT(NP2,NP4)=ASAT(NP2,NP4)-TEMP3
      ASAT(NP3,NP1)=ASAT(NP3,NP1)-TEMP1
      ASAT(NP3,NP2)=ASAT(NP3,NP2)-TEMP2
      ASAT(NP3,NP3)=ASAT(NP3,NP3)+TEMP1
      ASAT(NP3,NP4)=ASAT(NP3,NP4)+TEMP2
      ASAT(NP4,NP1)=ASAT(NP4,NP1)-TEMP2
      ASAT(NP4,NP2)=ASAT(NP4,NP2)-TEMP3
      ASAT(NP4,NP3)=ASAT(NP4,NP3)+TEMP2
   11 ASAT(NP4,NP4)=ASAT(NP4,NP4)+TEMP3
   12 DO 110 I=1,NTP
  110 INDEX(I)=0
  111 AMAX=-1.
      DO 114 I=1,NTP
      IF (INDEX(I)) 114,112,114
  112 TEMP=ABS(ASAT(I,I))
      IF (TEMP-AMAX) 114,114,113
  113 ICOL=I
      AMAX=TEMP
  114 CONTINUE
      IF (AMAX) 120,138,115
  115 INDEX (ICCL)=1
      PIVOT=ASAT(ICCL,ICOL)
      ASAT(ICOL,ICOL)=1.0
      PIVOT=1./PIVOT
      DO 116 J=1,NTP
  116 ASAT(ICOL,J)=ASAT(ICOL,J)*PIVOT
      DO 119 I=1,NTP
      IF (I-ICOL) 117,119,117
  117 TEMP=ASAT(I,ICOL)
      ASAT(I,ICCL)=0.0
      DO 118 J=1,NTP
  118 ASAT(I,J)=ASAT(I,J)-ASAT(ICOL,J)*TEMP
  119 CONTINUE
      GO TO 111
  120 CONTINUE
  612 PRINT 601
  601 FORMAT(≠1≠,≠  INFLUENCE COEFFIEIENTS≠//)
      DO 214 I=1,NTP
      PRINT 602,I
  602 FORMAT(≠ ROW NO.=≠,I5)
      PRINT 603,(ASAT(I,J),J=1,NTP)
  603 FORMAT(8E16.7)
```

```
  214 CONTINUE
  125 PRINT 126
  126 FORMAT (13H0THE MATRIX P)
      DO 127 I=1,NTP
  127 PRINT 128,I,(P(I,J),J=1,NLC)
  128 FORMAT (4H ROW,I3,1X,4(1PE16.7)/(8X,4(1PE16.7)))
      DO 129 I=1,NTP
      DO 129 J=1,NLC
      X(I,J)=0.
      DO 129 K=1,NTP
  129 X(I,J)=X(I,J)+ASAT(I,K)*P(K,J)
      PRINT 130
  130 FORMAT (13H0THE MATRIX X)
      DO 131 I=1,NTP
  131 PRINT 128,I,(X(I,J),J=1,NLC)
      PRINT 132
  132 FORMAT (49H1THE MEMBER FORCES IN ORDER OF LOADING CONDITIONS/)
      DO 133 J=1,NLC
  133 X(NPP1,J)=0.
      DO 134 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      CS=H/XL
      SN=V/XL
      DO 136 J=1,NLC
C
C     GENERATE ≠B≠ COLUMN FOR STRESS CONSTRAINTS
C
      F(J)=E*A/XL*(CS*(X(NP3,J)-X(NP1,J))+SN*(X(NP4,J)-X(NP2,J)))
      LL=NF*(J-1)+I
  540 BASE(LL,LL1)=-F(J)
  136 CONTINUE
  134 PRINT 137, MEMNO,(F(J),J=1,NLC)
  137 FORMAT (1H ,I5,5F12.4/(6X,5F12.4))
      PRINT 250
  250 FORMAT (53H0CHANGES IN BAR FORCES DUE TO SUCCESSIVE AREA CHANGES/)
      DO 201 I=1,NPP1
      ASAT(I,NPP1)=0.
  201 ASAT(NPP1,I)=0.
      DO 202 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
```

```
      XL=SQRT(H*H+V*V)
      CS=H/XL
      SN=V/XL
      NP(1)=NP1
      NP(2)=NP2
      NP(3)=NP3
      NP(4)=NP4
      SA(1)=-E*A*CS/XL
      SA(2)=-E*A*SN/XL
      SA(3)=E*A*CS/XL
      SA(4)=E*A*SN/XL
      DO 202 J=1,NTP
      SAK(MEMNO,J)=0.
      DO 202 K=1,4
      K1=NP(K)
  202 SAK(MEMNO,J)=SAK(MEMNO,J)+SA(K)*ASAT(K1,J)
      DO 203 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      CS=H/XL
      SN=V/XL
      EACOL=E*A*CS/XL
      EASOL=E*A*SN/XL
      TEMP1=EACOL*CS
      TEMP2=EACOL*SN
      TEMP3=EASOL*SN
      NP(1)=NP1
      NP(2)=NP2
      NP(3)=NP3
      NP(4)=NP4
      DO 213 J=1,NF
      IF (I-J) 205,206,205
  205 SA(1)=0.
      SA(2)=0.
      GO TO 207
  206 SA(1)=-EACOL
      SA(2)=-EASOL
  207 A=SAK(J,NP1)-SAK(J,NP3)
      B=SAK(J,NP2)-SAK(J,NP4)
      SA(1)=SA(1) - TEMP1*A - TEMP2*B
      SA(2)=SA(2) - TEMP2*A - TEMP3*B
      SA(3)=-SA(1)
      SA(4)=-SA(2)
      DO 204 K=1,NLC
      DELF(K)=0.
      DO 204 L=1,4
      L1=NP(L)
  204 DELF(K)=DELF(K)+SA(L)*X(L1,K)
```

```
            PRINT 137,MEMNO,(DELF(K),K=1,NLC)
      C
      C     GENERATE COEFFICIENTS FOR STRESS CONSTRAINTS.
      C
            DO 302 K=1,NLC
            LL=NF*(K-1)+J
        302 BASE(LL,I)=DELF(K)
        213 CONTINUE
        203 CONTINUE
            DO 217 J=1,NF
            DO 218 I=1,NPP1
            DO 218 K=1,NPP1
        218 ASB(I,K)=0.0
            PRINT 604,J
        604 FORMAT(≠0≠,≠CHANGES IN JOINT DEFORMATIONS DUE TO CHANGE IN THE ,
           1AREA OF MEMBER NO. ≠,I5/)
            I=J
            MEMNO=MNO(I)
            NP1=NPN(I,1)
            NP2=NPN(I,2)
            NP3=NPN(I,3)
            NP4=NPN(I,4)
            H=HVA(I,1)
            V=HVA(I,2)
            A=HVA(I,3)
            XL=SQRT(H*H+V*V)
            CS=H/XL
            SN=V/XL
            TEMP1=E*A*CS*CS/XL
            TEMP2=E*A*CS*SN/XL
            TEMP3=E*A*SN*SN/XL
            ASB(NP1,NP1)=ASB(NP1,NP1)+TEMP1
            ASB(NP1,NP2)=ASB(NP1,NP2)+TEMP2
            ASB(NP1,NP3)=ASB(NP1,NP3)-TEMP1
            ASB(NP1,NP4)=ASB(NP1,NP4)-TEMP2
            ASB(NP2,NP1)=ASB(NP2,NP1)+TEMP2
            ASB(NP2,NP2)=ASB(NP2,NP2)+TEMP3
            ASB(NP2,NP3)=ASB(NP2,NP3)-TEMP2
            ASB(NP2,NP4)=ASB(NP2,NP4)-TEMP3
            ASB(NP3,NP1)=ASB(NP3,NP1)-TEMP1
            ASB(NP3,NP2)=ASB(NP3,NP2)-TEMP2
            ASB(NP3,NP3)=ASB(NP3,NP3)+TEMP1
            ASB(NP3,NP4)=ASB(NP3,NP4)+TEMP2
            ASB(NP4,NP1)=ASB(NP4,NP1)-TEMP2
            ASB(NP4,NP2)=ASB(NP4,NP2)-TEMP3
            ASB(NP4,NP3)=ASB(NP4,NP3)+TEMP2
        219 ASB(NP4,NP4)=ASB(NP4,NP4)+TEMP3
      C
      C
            DO 220 I=1,NTP
            DO 220 L=1,NTP
            T=0.0
            DO 221 K=1,NTP
        221 T=T-ASAT(I,K)*ASB(K,L)
        220 C(I,L)=T
```

```
      C
      C
            DO 224 I=1,NTP
            DO 222 K=1,NLC
            T=0.0
            DO 223 M=1,NTP
        223 T=T+C(I,M)*X(M,K)
        222 DELX(K)=T
      C
      C     GENERATE SLACK VARIABLES FOR STRESS CONSTRAINTS.
      C
            M2=NF*NLC
            DO 503 I1=1,M2
            J1=NF+I1
            DO 511 K=1,NLC
            IF (-BASE(I1,LL1)) 510,520,520
        510 BASE (I1,J1)=-1.0
            GO TO 511
        520 BASE(I1,J1)=1.0
        511 CONTINUE
        503 CONTINUE
      C
      C     GENERATE COEFFICIENTS FER DEFORMATION CONSTRAINTS.
      C
            IF (I.EQ.NTP1)  GO TO 303
            IF(I.EQ.NTP2)  GO TO 304
            GO TO 224
        303 DO 305 K=1,NLC
            IF(X(NTP1,K)) 350,351,351
        350 BASE(NLC*NF+K,LL1)=-XALL-2*X(NTP1,K)
            GO TO 305
        351 BASE(NLC*NF+K,LL1)=XALL-2*X(NTP1,K)
        305 BASE(NLC*NF+K,J) = DELX(K)
            GO TO 224
        304 DO 306 K=1,NLC
            IF(X(NTP2,K)) 355,356,356
        355 BASE(NLC*NF+NLC+K,LL1)=-XALL-2*X(NTP2,K)
            GO TO 306
        356 BASE(NLC*NF+NLC+K,LL1)=XALL-2*X(NTP2,K)
        306 BASE(NLC*NF+NLC+K,J)=DELX(K)
      C
        224 PRINT 603,(DELX(K),K=1,NLC)
      C
      C
        217 CONTINUE
      C
      CGENERATE STRESSCONSTR. COEFF. ON DIAGONAL (DELF(I,I)- FALLOW(I,K*A(I))
      C
            DO 307 K=1,NLC
            DO 307 I=1,NF
            L2=NF*(K-1)+I
            IF(BASE(L2,LL1))900,900,910
        900 BASE(L2,I)=BASE(L2,I)-FALLT*HVA(I,3)
            GO TO 307
        910 BASE(L2,I) = BASE(L2,I)-FALLOW(I)*HVA(I,3)
```

```
  307 CONTINUE
C
C     MAKE ALL STRESS CONSTR. EQ. RIGHT HAND SIDE POSITIVE.
C
      M4=NF*NLC
      DO 450 J=1,LL1
      DO 450 I1=1,M4
      IF (BASE(I1,LL1)) 400,410,410
  400 BASE(I1,J)=-BASE(I1,J)
  410 CONTINUE
  450 CONTINUE
C
C     GENERATE SLACK VARIABLES FOR DEFORMATION CONSTRAINTS.
C
      DO 308 K=1,NLC
      J=NF+NF*NLC+K
      IF(X(NTP1,K)) 460,470,470
  460 BASE(NLC*NF+K,J)=-1.0
      GO TO 504
  470 BASE(NLC*NF+K,J)=1.0
  504 J1=NF+NF*NLC+NLC+K
      IF(X(NTP2,K)) 480,490,490
  480 BASE(NLC*NF+NLC+K,J1)=-1.0
      GO TO 308
  490 BASE(NLC*NF+NLC+K,J1)=1.0
  308 CONTINUE
C
C     MAKE ALL DEFORM. CONSTR. RIGHT HAND SIDE POSITIVE.
C
      M6=NF*NLC+1
      M7=NF*NLC+NCP*NLC
      DO 491 I=M6,M7
      DO 491 J=1,LL1
      IF(BASE(I,LL1)) 492,494,494
  492 BASE(I,J)=-BASE(I,J)
  494 CONTINUE
  491 CONTINUE
C
C     GENERATE SLACK VARIABLES FOR UB, LB, AMIN.
C
      DO 309 I=1,NF
      I1=NF*NLC+NCP*NLC+I
      J1=NF+NF*NLC+NCP*NLC+I
      BASE(I1,I)=1.0
      BASE(I1,J1)=1.0
      BASE(I1,LL1)=UB
      I2=I1+NF
      J2=J1+NF
      BASE(I2,I)=1.0
      BASE(I2,J2)=-1.0
      BASE(I2,LL1)=XLB
      I3=I1+NF*2
      J3=J1+NF*2
      BASE(I3,I)=HVA(I,3)
      BASE(I3,J3)=-1.0
```

```
          BASE(I3,LL1)=AMIN
    309   CONTINUE
          PRINT 502
    502   FORMAT(1H1,50X,35HCOEFFICIENTS FOR LP SIMPLEX TABLEAU//)
C
C         GENERATE COEFFICIENTS FOR  IDENTICAL AREA EQUATIONS.
C
          I2=NF*NLC+3*NF+NCP*NLC
          BASE(I2+1,1)  =  1.0
          BASE(I2+1,3)  =-1.0
          BASE(I2+2,4)  =  1.0
          BASE(I2+2,5)  =-1.0
          BASE(I2+3,8)  =  1.0
          BASE(I2+3,9)  =  -1.0
          DO 500 I=1,M1
    500   PRINT 501,((BASE(I,J),J=1,LL2),BASE(I,LL1))
    501   FORMAT (13F10.5)
    138   CALL LINPROG (BASE,M3,LL2,HVA,NF,KH,KHM)
          CALL DEMTR(NTP,NF,NLC,E,MNO,NPN,HVA,P,FALLOW,FALLT,XALL,
         1 NTP1,NTP2,NO,NAME,NA,NS,FMIN,FMAX,W,NL,NLMIN,CO,CP,XM,KC,KH,KHM,
         1FY,KCM)
          KH=KH+1
          IF (KH.GT.KHM) GO TO 541
          J=KC
          J1=KC-1
          T=ABS(W(J1)-W(J))/W(J1)
          IF(T.LT.0.005) GO TO 541
          GO TO 820
    541   RETURN
          END
```

A.4.3 LISTING OF SUBROUTINE LINPROG

```
      SUBROUTINE LINPROG(A,M,N,HVA,NF,KH,KHM)
C     LINEAR PROGRAMMING SIMPLEX TABLEAU
      DIMENSION A(85,175),IBAS(85),R(85)
      DIMENSION DES(85),HVA(21,3)
      PRINT 200,M,N
200   FORMAT (2I3)
      NM=N+M
      MZ=M+1
      MW=M+2
      MW1=MW
      NB=N+M+1
      I=0
      J=N
2     I=I+1
      J=J+1
      A(I,J)=1.0
      IF (J-NM) 2,3,3
3     DO 4 I=1,M
      A(MW,NB)=A(MW,NB)-A(I,NB)
      DO 4 J=1,N
      A(MW,J)=A(MW,J)-A(I,J)
4     CONTINUE
      DO 5 I=1,M
      IBAS(I)=N+I
5     CONTINUE
      PRINT 203, (IBAS(I),I=1,M)
203   FORMAT (26I3/)
      K=0
6     K=K+1
      I=1
      J=2
12    IF(A(MW,I)-A(MW,J)) 8,9,9
9     I=J
8     J=J+1
      IF (MW-MW1) 11,10,10
10    IF (J-NM) 12,12,13
11    IF (J-N) 12,12,13
13    IF (A(MW,I)+.0001) 16,17,17
16    IS=I
      PRINT 204,K,IS
204   FORMAT (2X,I3,17HCYCLE  INCOMING  ,I3)
      I=1
      J=0
18    IF (A(I,NB)) 19,20,20
19    IF (A(I,NB)-.0001) 23,20,20
23    PRINT 205,K,I,A(I,NB)
205   FORMAT (2I3,F10.2,11HB  NEGATIVE)
      GO TO 51
20    IF (A(I,IS)) 25,25,26
25    J=J+1
      R(I)=1000000.000
      GO TO 27
26    R(I)=A(I,NB)/A(I,IS)
27    I=I+1
      IF (I-M) 18,18,29
```

```
      29   IF (J-M) 31,30,30
      30   PRINT 206
      206  FORMAT (19HFUNCTION  UNBOUNDED)
           GO TO 100
      31   I=1
           J=2
      34   IF (R(I)-R(J)) 33,32,32
      32   I=J
      33   J=J+1
           IF (J-M) 34,34,35
      35   IR=I
           PRINT 207,K,IR
      207  FORMAT (2X,I3,17HCYCLE  OUTGOING  ,I3)
           IBAS(IR)=IS
           B2=A(IR,IS)
           DO 36 J=1,NB
           A(IR,J)=A(IR,J)/B2
      36   CONTINUE
           I=0
      37   I=I+1
           IF (I-IR) 38,37,38
      38   B=-A(I,IS)
           DO 39 J=1,NB
           A(I,J)=A(I,J)+B*A(IR,J)
      39   CONTINUE
           IF (I-MW) 37,41,41
      41   PRINT 208,K
      208  FORMAT (2X,I3,21HCYCLE  REDUCED  TABLO)
           GO TO 6
      17   IF (MW-MW1) 42,43,43
      43   IF (A(MW,NB)) 45,46,47
      45   B3=-A(MW,NB)
           GO TO 48
      47   B3=A(MW,NB)
      48   IF (B3-.0001) 46,46,49
      49   PRINT 210,K
      210  FORMAT (I3,12HW  NOT  ZERO)
           GO TO 51
      46   MW=M+1
           GO TO 6
      51   PRINT 211
      211  FORMAT (6X,20HINFEASABLE  SOLUTION)
           GO TO 100
      42   Z=-A(MZ,NB)
           DO 900 I=1,MZ
      900  PRINT 901,(A(I,J),J=1,NM),A(I,NB)
      901  FORMAT(1X,12F11.5)
           PRINT 212,Z
      212  FORMAT (6X,22HOPTIMUM  SOLUTION  Z= ,F10.2,//)
           PRINT 215,(IBAS(I),A(I,NB),I=1,M)
      215  FORMAT ((6X,2HX,I2,2H=,F10.2)/)
           DO 401 I=1,NF
           DO 402 J=1,M
           IF(IBAS(J).EQ.I) GO TO 403
           GO TO 402
```

```
      403  DES(I)=A(J,NB)
           GO TO 401
      402  CONTINUE
      401  CONTINUE
115        DO 404 I=1,NF
           PRINT 215,I,DES(I)
           HVA(I,3)=HVA(I,3)*DES(I)
           PRINT 216,I,HVA(I,3)
      216  FORMAT ((6X,1HA,I3,1H=,F10.2)/)
120   404  CONTINUE
           GO TO 101
      100  KH=KHM+1
      101  RETURN
           END
```

A.4.4 LISTING OF SUBROUTINE DEMTR

```
      SUBROUTINE DEMTR(NP,NF,NLC,E,MNO,NPN,HVA,P,FALLOW,FALLT,XALL,
     1 NTP1,NTP2,NO,NAME,NA,NS,FMIN,FMAX,W,NL,NLMIN,CO,CP,XM,KC,KH,KHM,
     1FY,KCM)
      DIMENSION ASAT(22,22),P(21,8),X(22,8),F(10),INDEX(21)
      DIMENSION MNO(21),NPN(21,4),HVA(21,3)
      DIMENSION      FALLOW(22),W(10),RX(22,8)
      DIMENSION NA(60,5),NS(28),FMIN(28),FMAX(28),NO(60),NAME(60,2)
      REAL NA
      IF(KH.GT.KHM) GO TO 138
      L3=NTP1
      L6=NTP2
  204 NPP1=NP+1
      DO 103 I=1,NPP1
      DO 103 J=1,NPP1
  103 ASAT(I,J)=0.
      PRINT 105
  105 FORMAT (27H0MEMBER  NP1  NP2  NP3  NP4, 7X, 1HH, 11X, 1HV, 11X,
     1 1HA, 11X, 1HL, 11X, 3HCOS, 12X, 3HSIN,12X,6HFALLOW,12X,2HNS/)
      WL=0.0
      DO 108 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      XCOS=H/XL
      XSIN=V/XL
      WL=WL+A*XL*0.284
      PRINT 107,MEMNO,NP1,NP2,NP3,NP4,H,V,A,XL,XCOS,XSIN,FALLOW(I)
     1,NS(I)
  107 FORMAT (1H0,I5,I6,3I5,4F12.4,3F15.8,5X,I5)
      TEMP1=E*A*XCOS*XCOS/XL
      TEMP2=E*A*XCOS*XSIN/XL
      TEMP3=E*A*XSIN*XSIN/XL
      ASAT(NP1,NP1)=ASAT(NP1,NP1)+TEMP1
      ASAT(NP1,NP2)=ASAT(NP1,NP2)+TEMP2
      ASAT(NP1,NP3)=ASAT(NP1,NP3)-TEMP1
      ASAT(NP1,NP4)=ASAT(NP1,NP4)-TEMP2
      ASAT(NP2,NP1)=ASAT(NP2,NP1)+TEMP2
      ASAT(NP2,NP2)=ASAT(NP2,NP2)+TEMP3
      ASAT(NP2,NP3)=ASAT(NP2,NP3)-TEMP2
      ASAT(NP2,NP4)=ASAT(NP2,NP4)-TEMP3
      ASAT(NP3,NP1)=ASAT(NP3,NP1)-TEMP1
      ASAT(NP3,NP2)=ASAT(NP3,NP2)-TEMP2
      ASAT(NP3,NP3)=ASAT(NP3,NP3)+TEMP1
      ASAT(NP3,NP4)=ASAT(NP3,NP4)+TEMP2
      ASAT(NP4,NP1)=ASAT(NP4,NP1)-TEMP2
      ASAT(NP4,NP2)=ASAT(NP4,NP2)-TEMP3
      ASAT(NP4,NP3)=ASAT(NP4,NP3)+TEMP2
  108 ASAT(NP4,NP4)=ASAT(NP4,NP4)+TEMP3
      PRINT 140,WL
```

```
 140  FORMAT(25X,5HWL = ,F10.4)
 215  DO 110 I=1,NP
  110 INDEX(I)=0
  111 AMAX=-1.
      DO 114 I=1,NP
      IF (INDEX(I)) 114,112,114
  112 TEMP=ABS(ASAT(I,I))
      IF (TEMP-AMAX) 114,114,113
 113  ICOL=I
      AMAX=TEMP
  114 CONTINUE
      IF (AMAX) 120,138,115
  115 INDEX (ICOL)=1
      PIVOT=ASAT(ICOL,ICOL)
      ASAT(ICOL,ICOL)=1.0
      PIVOT=1./PIVOT
      DO 116 J=1,NP
  116 ASAT(ICOL,J)=ASAT(ICOL,J)*PIVOT
      DO 119 I=1,NP
      IF (I-ICOL) 117,119,117
  117 TEMP=ASAT(I,ICOL)
      ASAT(I,ICOL)=0.0
      DO 118 J=1,NP
  118 ASAT(I,J)=ASAT(I,J)-ASAT(ICOL,J)*TEMP
  119 CONTINUE
      GO TO 111
 120  CONTINUE
  125 PRINT 126
  126 FORMAT (13H0THE MATRIX P)
      DO 127 I=1,NP
 127  PRINT 128,I,(P(I,J),J=1,NLC)
  128 FORMAT (4H ROW,I3,1X,4(1PE16.7)/(8X,4(1PE16.7)))
      DO 129 I=1,NP
      DO 129 J=1,NLC
      X(I,J)=0.
      DO 129 K=1,NP
  129 X(I,J)=X(I,J)+ASAT(I,K)*P(K,J)
      PRINT 130
  130 FORMAT (13H0THE MATRIX X)
      DO 131 I=1,NP
 131  PRINT 128,I,(X(I,J),J=1,NLC)
      DO 201 I=1,NP
      DO 200 J=1,NLC
      IF(X(I,J)) 260,261,261
 260  RX(I,J)=X(I,J)/(-XALL)
      GO TO 200
 261  RX(I,J)=X(I,J)/XALL
 200  CONTINUE
 201  CONTINUE
      DO 210 I=L3,L6,3
      PRINT 211,(RX(I,J),J=1,NLC)
 211  FORMAT(33X,F10.4)
 210  CONTINUE
      RXMAX=RX(L3,1)
      DO 302 I=L3,L6,3
```

```
      DO 302 J=1,NLC
      IF(RXMAX.LT.RX(I,J)) RXMAX=RX(I,J)
  302 CONTINUE
      PRINT 310,RXMAX
  310 FORMAT(25X,8HRXMAX = F10.4)
      PRINT 132
  132 FORMAT (49H1THE MEMBER FORCES IN ORDER OF LOADING CONDITIONS/)
      DO 133 J=1,NLC
  133 X(NPP1,J)=0.
      DO 134 I=1,NF
      MEMNO=MNO(I)
      NP1=NPN(I,1)
      NP2=NPN(I,2)
      NP3=NPN(I,3)
      NP4=NPN(I,4)
      H=HVA(I,1)
      V=HVA(I,2)
      A=HVA(I,3)
      XL=SQRT(H*H+V*V)
      XCOS=H/XL
      XSIN=V/XL
      FMIN(I)=0.0
      FMAX(I)=0.0
      DO 136 J=1,NLC
      F(J)=E*A/XL*(XCOS*(X(NP3,J)-X(NP1,J))+XSIN*(X(NP4,J)-X(NP2,J)))
  136 CONTINUE
      PRINT 137, MEMNO,(F(J),J=1,NLC)
  137 FORMAT (1H ,I5,5F12.4/(6X,5F12.4))
C
C     FIND MAX AND MIN MEMBER FORCES.
C
      DO 265 J=1,NLC
      IF(F(J)) 202,203,203
  202 IF (FMIN(I).GT.F(J)) FMIN(I)=F(J)
      GO TO 265
  203 IF(FMAX(I).LT.F(J)) FMAX(I) =F(J)
  265 CONTINUE
      PRINT 266 ,MEMNO    ,FMIN(I),FMAX(I),XL
  266 FORMAT(1H ,I5,3F12.4)
C
C     FIND SMALLEST SECTION FOR TENSION.
C
      AT=FMAX(I)/FALLT
      I1=NLMIN
      IF (ABS(AT).LT.0.0001) GO TO 236
  235 IF(NA(I1,1).LT.AT) GO TO 234
  404 IF(NA(I1,1).LT.HVA(I,3)*RXMAX)GO TO 234
  405 IF(NA(I1,2).LE.NA(I1,3))RI1=NA(I1,2)
      IF(NA(I1,2).GT.NA(I1,3))RI1=NA(I1,3)
      IF   (RI1.LT.XL/300.0) GO TO 234
      IF (ABS(FMIN(I)).LT.0.0001) GO TO 270
      GO TO 236
  234 I1=I1+1
      IF(I1.GT.NL) GO TO 240
      GO TO 235
```

```
C
C      FIND SMALLEST SECTION FOR COMPRESSION AND TEST FOR LOCAL BUCKLING.
C
  236  I2=I1
  237  IF(NA(I2,2).LE.NA(I2,3))RI2=NA(I2,2)
       IF(NA(I2,2).GT.NA(I2,3))RI2= NA(I2,3)
       IF   (RI2.LT.XL/200.0) GO TO 238
       FY=44.00
       B=NA(I2,4)
       T=NA(I2,5)
       IF(B/T.GT.(75.0/SQRT(FY)))FY=1.67*(FY-8.6)-0.067*(FY-21.6)*B/T
C
C      COMPUTE ALLOW. COMPRESSION STRESS FOR EVERY SECTION UNTIL A SATISFACTO
C      ONE IS FOUND.
       IF(XL/RI2.LE.CO) FAI2=0.60*FY
       IF(XL/RI2.GT.CO.AND.XL/RI2.LE.CP) FAI2=0.60*FY-
      1XM*(XL/RI2-CO)
       IF(XL/RI2.GT.CP) FAI2=149000.0/((XL/RI2)*(XL/RI2))
       IF(ABS(FAI2*NA(I2,1)) .LT.ABS(FMIN(I))) GO TO 238
  406  IF(NA(I2,1).LT.HVA(I,3)*RXMAX)GO TO 238
       GO TO 239
  238  I2=I2+1
       IF(I2.GT.NL) GO TO 240
       GO TO 237
  240  NS(I)=I1
       HVA(I,3)=NA(I1,1)
       FALLOW(I)=-0.1
       GO TO 134
  239  NS(I)=I2
       HVA(I,3)=NA(I2,1)
       FALLOW(I)=-FAI2
  134  CONTINUE
       IF (HVA(1,3).LT.HVA(3,3)) GO TO 300
       IF(HVA(1,3).GE.HVA(3,3)) GO TO 301
  307  IF (HVA(4,3).LT.HVA(5,3))  GO TO 332
       IF (HVA(4,3).GE.HVA(5,3)) GO TO 303
  308  IF (HVA(8,3).LT.HVA(9,3))  GO TO 304
       IF(HVA(8,3).GE.HVA(9,3))  GO TO 305
       GO TO 309
  300  HVA(1,3)=HVA(3,3)
       NS(1)=NS(3)
       FALLOW(1)=FALLOW(3)
       GO TO 307
  301  HVA(3,3)=HVA(1,3)
       NS(3)=NS(1)
       FALLOW(3)=FALLOW(1)
       GO TO 307
  332  HVA(4,3)=HVA(5,3)
       NS(4)=NS(5)
       FALLOW(4)=FALLOW(5)
       GO TO 308
  303  HVA(5,3)=HVA(4,3)
       NS(5)=NS(4)
       FALLOW(5)=FALLOW(4)
       GO TO 308
```

```
304  HVA(8,3)=HVA(9,3)
     NS(8)=NS(9)
     FALLOW(8)=FALLOW(9)
     GO TO 309
305  HVA(9,3)=HVA(8,3)
     NS(9)=NS(8)
     FALLOW(9)=FALLOW(8)
309  CONTINUE
     PRINT 280
280  FORMAT (27H0MEMBER  NP1  NP2  NP3  NP4, 7X, 1HH, 11X,1HV,11X,
    1  1HA, 11X, 1HL, 11X, 3HCOS, 12X, 3HSIN,12X,6HFALLOW,12X,2HNS/)
     KC=KC+1
     J=KC
     W(J)=0.0
     DO 350 I=1,NF
     MEMNO=MNO(I)
     NP1=NPN(I,1)
     NP2=NPN(I,2)
     NP3=NPN(I,3)
     NP4=NPN(I,4)
     H=HVA(I,1)
     V=HVA(I,2)
     A=HVA(I,3)
     XL=SQRT(H*H+V*V)
     XCOS=H/XL
     XSIN=V/XL
     W(J)=W(J)+A*XL*0.284
     PRINT 281,MEMNO,NP1,NP2,NP3,NP4,H,V,A,XL,XCOS,XSIN,FALLOW(I),NS(I)
281  FORMAT (1H0,I5,I6,3I5,4F12.4,3F15.8,5X,I5)
350  CONTINUE
     PRINT 320,KH,J,W(J)
320  FORMAT(25X,4HKH =,I3,3X,7HWEIGHT ,I3,3H = ,F10.4)
     GO TO 138
240  PRINT 241
241  FORMAT(41H0INFEASIBLE WITH PRESENT LIST OF SECTIONS)
     KH=KHM+1
138  RETURN
     END
```